


DATA TRANSFER CONTROLLER, ELECTRONIC APPARATUS, AND DATA TRANSFER CONTROL METHOD


Publication number: JP2004021613 (A)

Also published as:

Publication date: 2004-01-22

 US2004073697 (A1)

Inventor(s): SAITO NOBUYUKI; SHIMONO HIROAKI; ISHIDA TAKUYA; KANBARA YOSHIYUKI; NAGAO KANEAKI +

 US7349973 (B2)

Applicant(s): SEIKO EPSON CORP +

Classification:

- **International:** G06F13/12; G06F13/36; G06F13/38; G06F13/42; G06F15/16; G06F13/12; G06F13/36; G06F13/38; G06F13/42; G06F15/16; (IPC1-7): G06F13/12; G06F13/36; G06F13/38

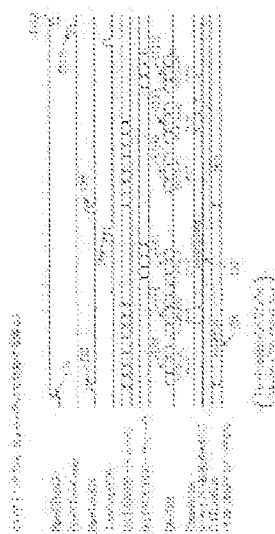
- **European:** G06F13/4256

Application number: JP20020175775 20020617

Priority number(s): JP20020175775 20020617

Abstract of JP 2004021613 (A)

PROBLEM TO BE SOLVED: To provide a data transfer controller, an electronic apparatus and a data transfer control method, by which the processing load of a processing part can be reduced. ; **SOLUTION:** Transaction corresponding to an end point is automatically generated, and data are automatically transferred while the residual data size of the transfer data is calculated based on the total size and maximum packet size. When the residual data size in this time transaction is less than the maximum packet size, the next transaction is automatically generated, and a short packet is automatically transferred to an end point. When the payload size of a packet to be transferred in this time transaction is the maximum packet size, and the residual data size of the transfer data is zero, a short packet whose data length is zero is automatically transferred to the end point. When direct memory access(DMA) transfer is completed, and the residual data to be transferred are zero, the short packet whose data length is zero is automatically transferred from a host to an IN token. The On-The-Go data transfer of a universal serial bus(USB) is carried out. ; **COPYRIGHT:** (C) 2004,JPO



~~~~~  
Data supplied from the **espacenet** database — Worldwide

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-21613

(P2004-21613A)

(43) 公開日 平成16年1月22日(2004.1.22)

(51) Int.Cl.<sup>7</sup>

F I

テーマコード(参考)

G06F 13/12

G06F 13/12 310C

5B014

G06F 13/36

G06F 13/12 330F

5B061

G06F 13/38

G06F 13/36 310F

5B077

G06F 13/38 310D

G06F 13/38 350

審査請求 有 請求項の数 13 O L (全 40 頁)

(21) 出願番号 特願2002-175775(P2002-175775)

(22) 出願日 平成14年6月17日(2002.6.17)

特許法第30条第1項適用申請有り 2002年4月1日 CQ出版株式会社発行の「Design Wave 2002年4月号」に発表

(71) 出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(74) 代理人 100090479

弁理士 井上 一

(74) 代理人 100090387

弁理士 布施 行夫

(74) 代理人 100090398

弁理士 大淵 美千栄

(72) 発明者 ▲斎▼藤 伸之

長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

(72) 発明者 下野 洋昭

長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

最終頁に続く

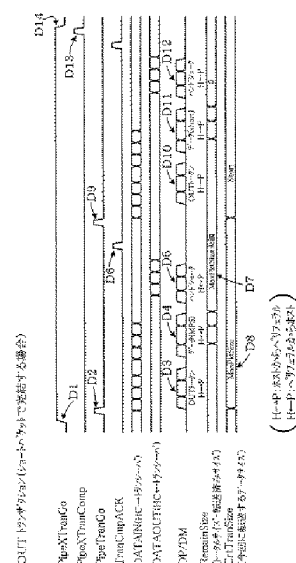
(54) 【発明の名称】 データ転送制御装置、電子機器及びデータ転送制御方法

(57) 【要約】

【課題】処理部の処理負荷の軽減等を図れるデータ転送制御装置、電子機器及びデータ転送制御方法を提供すること。

【解決手段】エンドポイントに対するトランザクションを自動発生し、トータルサイズとマックスパケットサイズに基づき転送データの残りデータサイズを演算しながらデータを自動転送する。今回のトランザクションでの残りデータサイズがマックスパケットサイズ未満の場合には、次のトランザクションを自動発生し、ショートパケットをエンドポイントに自動転送する。今回のトランザクションで転送するパケットのペイロードサイズがマックスパケットサイズであり、転送データの残りデータサイズが零の場合には、データ長が零のショートパケットをエンドポイントに自動転送する。DMA転送が完了し、転送すべき残りデータが零の場合には、ホストからのINTトークンに対してデータ長が零のショートパケットを自動転送する。USBのOn-The-Goのデータ転送を行う。

【選択図】 図7



**【特許請求の範囲】****【請求項1】**

バスを介したデータ転送のためのデータ転送制御装置であって、  
転送データを記憶するバッファのアクセス制御を行うバッファコントローラと、  
転送データのトータルサイズ、マックスパケットサイズが設定されるレジスタを含むレジスタ部と、  
エンドポイントに対するトランザクションを自動発生し、トータルサイズとマックスパケットサイズに基づき転送データの残りデータサイズを演算しながら、エンドポイントにデータを自動転送する転送コントローラとを含み、  
前記転送コントローラが、  
今回のトランザクションでの転送データの残りデータサイズがマックスパケットサイズ未満の場合には、次のトランザクションを自動発生して、ショートパケットをエンドポイントに自動転送することを特徴とするデータ転送制御装置。

**【請求項2】**

請求項1において、  
前記転送コントローラが、  
今回のトランザクションで転送するパケットのペイロードサイズがマックスパケットサイズであり、且つ、転送データの残りデータサイズが零の場合には、次のトランザクションを自動発生して、データ長が零のショートパケットをエンドポイントに自動転送することを特徴とするデータ転送制御装置。

**【請求項3】**

請求項1又は2において、  
前記バッファに、  
各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が確保され、  
前記レジスタ部が、  
各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報が各転送条件レジスタに設定される複数の転送条件レジスタを含み、  
前記トータルサイズと前記マックスパケットサイズが、転送条件情報として転送条件レジスタに設定され、  
前記転送コントローラが、  
転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを自動転送することを特徴とするデータ転送制御装置。

**【請求項4】**

請求項3において、  
前記転送コントローラが、  
各パイプ領域の転送データの残りデータサイズに基づいて、ショートパケットを転送するか否かの判断処理を行うことを特徴とするデータ転送制御装置。

**【請求項5】**

請求項1乃至4のいずれかにおいて、  
ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、  
前記転送コントローラが、  
ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、  
ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコントローラとを含み、  
ホスト動作時には、

各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が、パケットバッファに確保され、前記ホストコントローラが、パイプ領域とそのパイプ領域に対応するエンドポイントとの間でデータを転送し、

ペリフェラル動作時には、

ホストとの間で転送されるデータが各エンドポイント領域に記憶される複数のエンドポイント領域が、パケットバッファに確保され、前記ペリフェラルコントローラが、エンドポイント領域とホストとの間でデータを転送し、

前記ホストコントローラが、

ホスト動作時に、パイプ領域に対応するエンドポイントにショートパケットを自動転送することを特徴とするデータ転送制御装置。

【請求項6】

請求項1乃至5のいずれかにおいて、

前記バスとは異なる他のバスとパケットバッファとの間でのデータ転送を行うインターフェース回路を含み、

前記インターフェース回路、前記転送コントローラに対して、処理部がデータ転送の開始を指示した場合に、前記インターフェース回路が他のバスを介したデータ転送を行うと共に、前記転送コントローラがバスを介したデータ転送を行い、データ転送が終了した場合に、前記転送コントローラが、処理部に対して割り込みを発生することを特徴とするデータ転送制御装置。

【請求項7】

バスを介したデータ転送のためのデータ転送制御装置であって、

転送データを記憶するパケットバッファのアクセス制御を行うバッファコントローラと、前記バスとは異なる他のバスとパケットバッファとの間で、DMA (Direct Memory Access) 転送を行うインターフェース回路と、

ホストからINトークンを受信した場合に、ホストにデータを自動転送する転送コントローラとを含み、

前記転送コントローラが、

他のバスとパケットバッファとの間でのDMA転送が完了しており、且つ、転送すべき残りデータが零の場合には、ホストからのINトークンに対して、データ長が零のショートパケットをホストに自動転送することを特徴とするデータ転送制御装置。

【請求項8】

請求項7において、

ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、

前記転送コントローラが、

ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、

ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコントローラとを含み、

ホスト動作時には、

各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が、パケットバッファに確保され、前記ホストコントローラが、パイプ領域とそのパイプ領域に対応するエンドポイントとの間でデータを転送し、

ペリフェラル動作時には、

ホストとの間で転送されるデータが各エンドポイント領域に記憶される複数のエンドポイント領域が、パケットバッファに確保され、前記ペリフェラルコントローラが、エンドポイント領域とホストとの間でデータを転送し、

前記ペリフェラルコントローラが、

ペリフェラル動作時に、データ長が零のショートパケットをホストに自動転送することを特徴とするデータ転送制御装置。

## 【請求項9】

請求項8において、  
前記転送コントローラが、  
各エンドポイント領域の転送データの残りデータサイズに基づいて、データ長が零のショートパケットを転送するか否かの判断処理を行うことを特徴とするデータ転送制御装置。

## 【請求項10】

請求項1乃至9のいずれかにおいて、  
USB(Universal Serial Bus)のOTG(On-The-Go)規格に準拠したデータ転送を行うことを特徴とするデータ転送制御装置。

## 【請求項11】

請求項1乃至10のいずれかのデータ転送制御装置と、  
前記データ転送制御装置及びバスを介して転送されるデータの出力処理又は取り込み処理又は記憶処理を行う装置と、  
前記データ転送制御装置のデータ転送を制御する処理部と、  
を含むことを特徴とする電子機器。

## 【請求項12】

バスを介したデータ転送のためのデータ転送制御方法であって、  
転送データを記憶するパケットバッファのアクセス制御を行い、  
転送データのトータルサイズ、マックスパケットサイズをレジスタ部のレジスタに設定し、  
エンドポイントに対するトランザクションを自動発生し、トータルサイズとマックスパケットサイズに基づき転送データの残りデータサイズを演算しながら、エンドポイントにデータを自動転送すると共に、  
今回のトランザクションでの転送データの残りデータサイズがマックスパケットサイズ未満の場合には、次のトランザクションを自動発生して、ショートパケットをエンドポイントに自動転送することを特徴とするデータ転送制方法。

## 【請求項13】

バスを介したデータ転送のためのデータ転送制御方法であって、  
転送データを記憶するパケットバッファのアクセス制御を行い、  
前記バスとは異なる他のバスとパケットバッファとの間で、DMA(Direct Memory Access)転送を行い、  
ホストからINTトークンを受信した場合に、ホストにデータを自動転送すると共に、  
他のバスとパケットバッファとの間でのDMA転送が完了しており、且つ、転送すべき残りデータが零の場合には、ホストからのINTトークンに対して、データ長が零のショートパケットをホストに自動転送することを特徴とするデータ転送制御方法。

## 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明は、データ転送制御装置、電子機器及びデータ転送制御方法に関する。

## 【0002】

## 【背景技術及び発明が解決しようとする課題】

USB(Universal Serial Bus)2.0の市場が順調に拡大しているなか、USB Implementers Forum(USB-IF)によって、USB On-The-Go(OTG)と呼ばれるインターフェース規格が策定された。USB2.0を拡張する形で策定されたOTG規格(OTG1.0)は、USBインターフェースの新たな付加価値を生む可能性を秘めており、その特質を生かしたアプリケーションの登場が待たれている。

## 【0003】

このOTGによれば、これまでUSBを介してホスト(パーソナルコンピュータ等)に接続されていたペリフェラル(周辺機器)に、ホスト機能を持たせることができる。これにより、ペリフェラル同士をUSBで接続してデータを転送することが可能になり、例えば

デジタルカメラとプリンタとをダイレクトに接続して、デジタルカメラの画像を印刷することが可能になる。また、デジタルカメラやデジタルビデオカメラをストレージ装置に接続して、データを保存することが可能になる。

【0004】

しかしながら、OTGによりホスト機能を持たせるペリフェラルには、低性能のCPU（処理部）が組み込まれているのが一般的である。従って、ホスト機能の追加により、ペリフェラルが有するCPU（ファームウェア）の処理負荷が重くなったり、処理が複雑化すると、他の処理に支障が生じたり、機器の設計期間が長期化するなどの問題が生じる。

【0005】

特に、最後のトランザクションで転送されるパケットがショートパケットか否かをファームウェア（処理部）が判断すると、ファームウェアの処理負荷が重くなり、一連のパケットを自動転送することが困難になる。

【0006】

また、最後のトランザクションで転送されるパケットがマックスパケットサイズのパケットである場合に、転送データの終端を認識できない機器があることも判明した。

【0007】

本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、処理部の処理負荷の軽減等を図れるデータ転送制御装置、電子機器及びデータ転送制御方法を提供することにある。

【0008】

【課題を解決するための手段】

本発明は、バスを介したデータ転送のためのデータ転送制御装置であって、転送データを記憶するパケットバッファのアクセス制御を行うバッファコントローラと、転送データのトータルサイズ、マックスパケットサイズが設定されるレジスタを含むレジスタ部と、エンドポイントに対するトランザクションを自動発生し、トータルサイズとマックスパケットサイズに基づき転送データの残りデータサイズを演算しながら、エンドポイントにデータを自動転送する転送コントローラとを含み、前記転送コントローラが、今回のトランザクションでの転送データの残りデータサイズがマックスパケットサイズ未満の場合には、次のトランザクションを自動発生して、ショートパケットをエンドポイントに自動転送するデータ転送制御装置に関する。

【0009】

本発明では、転送データの残りデータサイズを演算しながら、エンドポイントにデータ（一連のパケット）が自動転送される。そして、転送データの残りデータサイズがマックスパケットサイズ未満になると、次のトランザクションが自動発生して、ショートパケットがエンドポイントに自動転送される。このようにすれば、最後のパケットがショートパケットか否かを処理部が判断すること無く、一連のパケットを用いて転送データをエンドポイントに自動転送できるようになる。これにより、処理部の処理負荷の軽減等を図れる。

【0010】

また本発明では、前記転送コントローラが、今回のトランザクションで転送するパケットのペイロードサイズがマックスパケットサイズであり、且つ、転送データの残りデータサイズが零の場合には、次のトランザクションを自動発生して、データ長が零のショートパケットをエンドポイントに自動転送するようにしてもよい。

【0011】

このようにすれば、転送相手となるペリフェラルが転送データの終端を認識できないような場合にも、零長のショートパケットを転送することで、これを認識させることが可能になる。

【0012】

また本発明では、前記パケットバッファに、各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が確保され、前記レジスタ部が、各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報が各転送条件レジスタに設

定される複数の転送条件レジスタを含み、前記トータルサイズと前記マックスパケットサイズが、転送条件情報として転送条件レジスタに設定され、前記転送コントローラが、転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを自動転送するようにしてもよい。

【0013】

本発明では、パケットバッファに対して複数のパイプ領域（バッファ領域）が確保（allocate）される。この場合に、各パイプ領域は、バスの各エンドポイントに対応して確保される。また、各パイプ領域には、それに対応する各エンドポイントとの間で転送（送信、受信）されるデータが記憶（バッファリング）される。バッファコントローラは、このようなパイプ領域が確保されたパケットバッファ（バッファ）のアクセス制御（領域管理）を行う。

【0014】

そして本発明では、各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報（エンドポイント情報、パイプ情報）が各転送条件レジスタ（パイプレジスタ）に設定される。そして、転送コントローラは、この転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生し、パイプ領域とエンドポイントとの間でデータを自動転送する。これにより、処理部の処理負荷を軽減できる。

【0015】

また本発明では、前記転送コントローラが、各パイプ領域の転送データの残りデータサイズに基づいて、ショートパケットを転送するか否かの判断処理を行うようにしてもよい。

【0016】

このようにすれば、各パイプ領域毎に、適切なショートパケット転送処理を行うことが可能になる。

【0017】

また本発明では、ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、前記転送コントローラが、ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコントローラとを含み、ホスト動作時には、各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が、パケットバッファに確保され、前記ホストコントローラが、パイプ領域とそのパイプ領域に対応するエンドポイントとの間でデータを転送し、ペリフェラル動作時には、ホストとの間で転送されるデータが各エンドポイント領域に記憶される複数のエンドポイント領域が、パケットバッファに確保され、前記ペリフェラルコントローラが、エンドポイント領域とホストとの間でデータを転送し、前記ホストコントローラが、ホスト動作時に、パイプ領域に対応するエンドポイントにショートパケットを自動転送するようにしてもよい。

【0018】

本発明によれば、例えば、ステートコントローラにより制御されるステートが、ホスト動作のステートになると、ホストコントローラにより、ホストの役割としてのデータ転送が行われる。また、ステートコントローラにより制御されるステートが、ペリフェラル動作のステートになると、ペリフェラルコントローラにより、ペリフェラルの役割としてのデータ転送が行われる。これにより、いわゆるデュアルロール・デバイス機能を実現できるようになる。そして本発明では、このようなデュアルロール・デバイス機能を実現できると共に、ホスト動作時における処理部の処理負荷を軽減できる。

【0019】

また本発明では、前記バスとは異なる他のバスとパケットバッファとの間でのデータ転送を行うインターフェース回路を含み、前記インターフェース回路、前記転送コントローラ

に対して、処理部がデータ転送の開始を指示した場合に、前記インターフェース回路が他のバスを介したデータ転送を行うと共に、前記転送コントローラがバスを介したデータ転送を行い、データ転送が終了した場合に、前記転送コントローラが、処理部に対して割り込みを発生するようにしてもよい。

【0020】

このようにすれば、処理部は、バス及び他のバス（処理部のバス又はシステムメモリのバス等）を介したデータ転送の開始を指示した後は、割り込みが発生するまで、データ転送制御装置の制御に関わらなくて済むようになる。これにより、処理部の処理負荷の軽減等を図れる。

【0021】

また本発明は、バスを介したデータ転送のためのデータ転送制御装置であって、転送データを記憶するバッファのアクセス制御を行うバッファコントローラと、前記バスとは異なる他のバスとバッファとの間で、DMA（Direct Memory Access）転送を行うインターフェース回路と、ホストからINTトークンを受信した場合に、ホストにデータを自動転送する転送コントローラとを含み、前記転送コントローラが、他のバスとバッファとの間でのDMA転送が完了しており、且つ、転送すべき残りデータが零の場合には、ホストからのINTトークンに対して、データ長が零のショートパケットをホストに自動転送するデータ転送制御装置に係る。

【0022】

本発明では、DMA転送が完了しており、且つ、転送すべき残りデータが零になると、次のトランザクションが自動発生して、データ長が零のショートパケットがホストに自動転送される。

【0023】

このようにすれば、転送相手となるホストが転送データの終端を認識できないような場合にも、零長のショートパケットを転送することで、これを認識させることが可能になる。

【0024】

また本発明では、ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、前記転送コントローラが、ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコントローラとを含み、ホスト動作時には、各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域が、バッファに確保され、前記ホストコントローラが、パイプ領域とそのパイプ領域に対応するエンドポイントとの間でデータを転送し、ペリフェラル動作時には、ホストとの間で転送されるデータが各エンドポイント領域に記憶される複数のエンドポイント領域が、バッファに確保され、前記ペリフェラルコントローラが、エンドポイント領域とホストとの間でデータを転送し、前記ペリフェラルコントローラが、ペリフェラル動作時に、データ長が零のショートパケットをホストに自動転送するようにしてもよい。

【0025】

また本発明では、前記転送コントローラが、各エンドポイント領域の転送データの残りデータサイズに基づいて、データ長が零のショートパケットを転送するか否かの判断処理を行うようにしてもよい。

【0026】

また本発明では、USB（Universal Serial Bus）のOTG（On-The-Go）規格に準拠したデータ転送を行うようにしてもよい。

【0027】

また本発明は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して転送されるデータの出力処理又は取り込み処理又は記憶処理を行う装置と、前記データ転送制御装置のデータ転送を制御する処理部とを含む電子機器に係る。



## 【0028】

また本発明は、バスを介したデータ転送のためのデータ転送制御方法であって、転送データを記憶するバッファのアクセス制御を行い、転送データのトータルサイズ、マックスパケットサイズをレジスタ部のレジスタに設定し、エンドポイントに対するトランザクションを自動発生し、トータルサイズとマックスパケットサイズに基づき転送データの残りデータサイズを演算しながら、エンドポイントにデータを自動転送すると共に、今回のトランザクションでの転送データの残りデータサイズがマックスパケットサイズ未満の場合には、次のトランザクションを自動発生して、ショートパケットをエンドポイントに自動転送するデータ転送制方法に関係する。

## 【0029】

また本発明は、バスを介したデータ転送のためのデータ転送制御方法であって、転送データを記憶するバッファのアクセス制御を行い、前記バスとは異なる他のバスとバッファとの間で、DMA (Direct Memory Access) 転送を行い、ホストからINTトークンを受信した場合に、ホストにデータを自動転送すると共に、他のバスとバッファとの間でDMA転送が完了しており、且つ、転送すべき残りデータが零の場合には、ホストからのINTトークンに対して、データ長が零のショートパケットをホストに自動転送するデータ転送制御方法に関係する。

## 【0030】

## 【発明の実施の形態】

以下、本実施形態について説明する。

## 【0031】

なお、以下に説明する本実施形態は、特許請求の範囲に記載された本発明の内容を不当に限定するものではない。また本実施形態で説明される構成の全てが本発明の解決手段として必須であるとは限らない。

## 【0032】

## 1. OTG

まず、OTG (USB On-The-Go) について簡単に説明する。

## 【0033】

## 1. 1 Aデバイス、Bデバイス

OTGでは、コネクタの規格として、図1 (A) に示すようなMini-Aプラグ、Mini-Bプラグが定義されている。また、これらのMini-Aプラグ、Mini-Bプラグ (広義にはケーブルの第1、第2のプラグ) の両方を接続できるコネクタとして、Mini-ABレセプタクル (receptacle) が定義されている。

## 【0034】

そして例えば図1 (B) のように、USBケーブルのMini-Aプラグに電子機器Pが接続され、Mini-Bプラグに電子機器Qが接続されると、電子機器PはAデバイスに設定され、電子機器QはBデバイスに設定される。一方、図1 (C) に示すように、Mini-Bプラグ、Mini-Aプラグが電子機器P、Qに接続されると、電子機器P、Qは、各々、Bデバイス、Aデバイスに設定される。

## 【0035】

Mini-Aプラグ内ではIDピンがGNDに接続されており、Mini-Bプラグ内ではIDピンはフローティング状態になっている。電子機器は、内蔵するプルアップ抵抗回路を用いて、このIDピンの電圧レベルを検出することで、自身がMini-Aプラグに接続されたのか、或いはMini-Bプラグに接続されたのかを判断する。

## 【0036】

OTGでは、Aデバイス (マスター) が、電源 (VBUS) を供給する側 (供給元) になり、Bデバイス (スレーブ) が、電源の供給を受ける側 (供給先) になる。また、Aデバイスは、デフォルトのステートがホストになり、Bデバイスは、デフォルトのステートがペリフェラル (周辺機器) になる。

## 【0037】

## 1. 2 デュアルロール・デバイス

OTGでは、ホスト（簡易ホスト）としての役割とペリフェラルとしての役割の両方を持つことができるデュアルロール・デバイス（Dual-Role Device）が定義されている。

### 【0038】

デュアルロール・デバイスは、ホストにもペリフェラルにもなれる。そして、デュアルロール・デバイスに接続された相手が、従来のUSB規格におけるホストやペリフェラルである場合には、デュアルロール・デバイスの役割は一意に定まる。つまり、接続相手がホストであれば、デュアルロール・デバイスはペリフェラルになり、接続相手がペリフェラルであれば、デュアルロール・デバイスはホストになる。

### 【0039】

一方、接続相手がデュアルロール・デバイスである場合には、両方のデュアルロール・デバイスは、お互いにホストとペリフェラルの役割を交換できる。

### 【0040】

## 1. 3 SRP、HNP

デュアルロール・デバイスは、図2（A）、（B）に示すようなセッション開始要求手順SRP（Session Request Protocol）やホスト交換手順HNP（Host Negotiation Protocol）の機能を持つ。

### 【0041】

ここでセッション開始要求手順SRPは、BデバイスがAデバイスに対して、VBUS（電源）の供給を要求するプロトコルである。

### 【0042】

バスを使用しない場合にOTGでは、AデバイスはVBUSの供給を停止できる。これにより、Aデバイスが例えば小型の携帯機器であった場合に、無駄な電力消費を防止できる。そして、AデバイスがVBUSの供給を停止した後に、BデバイスがVBUSを供給してもらいたい場合には、このSRPを使用して、Aデバイスに対してVBUSの供給の再開を要求する。

### 【0043】

図2（A）にSRPの流れを示す。図2（A）に示すように、Bデバイスは、データ・ライン・パルシングとVBUSパルシングを行うことで、Aデバイスに対してVBUSの供給を要求する。そして、AデバイスによるVBUSの供給の開始後に、Bデバイスのペリフェラル動作（peripheral operation）と、Aデバイスのホスト動作（host operation）が開始する。

### 【0044】

図1（A）～図1（C）で説明したように、デュアルロール・デバイス同士の接続では、Mini-Aプラグが接続された側であるAデバイスがデフォルトのホストとなり、Mini-Bプラグが接続された側であるBデバイスがデフォルトのペリフェラルになる。そして、OTGでは、プラグの抜き差しを行わなくても、ホストとペリフェラルの役割を交換できる。HNPは、このホストとペリフェラルの役割を交換するためのプロトコルである。

### 【0045】

HNPの流れを図2（B）に示す。デフォルトのホストとして動作するAデバイスが、バスの使用を終了すると、バスがアイドル状態になる。その後に、Bデバイスがデータ信号線DP（D+）のプルアップ抵抗を無効にすると、AデバイスがDPのプルアップ抵抗を有効にする。これにより、Aデバイスの役割はホストからペリフェラルに変わり、ペリフェラルとしての動作を開始する。また、Bデバイスの役割はペリフェラルからホストに変わり、ホストとしての動作を開始する。

### 【0046】

その後、Bデバイスが、バスの使用を終了し、AデバイスがDPのプルアップ抵抗を無効

にすると、Bデバイスが、DPのプルアップ抵抗を有効にする。これにより、Bデバイスの役割はホストからペリフェラルに戻り、ペリフェラルとしての動作を再開する。また、Aデバイスの役割はペリフェラルからホストに戻り、ホストとしての動作を再開する。

【0047】

以上に説明したOTGによれば、携帯電話やデジタルカメラなどの携帯機器をUSBのホストとして動作させ、携帯機器同士をピア・ツー・ピアで接続してデータ転送を行うことが可能になる。これにより、USBインターフェースに新たな付加価値を生むことができ、これまでには存在しなかったアプリケーションを創出できる。

【0048】

## 2. OHCI

さて、従来のUSBでは、ホストであるパーソナルコンピュータが有するデータ転送制御装置（ホストコントローラ）は、マイクロソフト社が提唱したOHCI（Open Host Controller Interface）や、UHCI（Universal Host Controller Interface）といった規格に準拠していた。また、使用されるOS（Operating System）も、マイクロソフト社のWindows（登録商標）やアップル社のマッキントッシュのOSなどに限定されていた。

【0049】

しかしながら、OTGのターゲット・アプリケーションである小型携帯機器では、組み込まれるCPUのアーキテクチャや、使用されるOSは千差万別である。更に、パーソナルコンピュータのホストコントローラ向けに規格化されたOHCIやUHCIは、USBホストとしての機能をフルに実装することを前提としており、小型携帯機器への実装に最適であるとは言い難い。

【0050】

例えば、図3（A）に、OHCIで使用されるリスト構造のディスクリプタの一例を示す。

【0051】

図3（A）において、エンドポイントディスクリプタED1、ED2、ED3は、リンクポイントによりリンクされており、エンドポイント1、2、3とのコミュニケーションに必要な情報が含まれている。そして、これらのED1、ED2、ED3には、転送ディスクリプタTD11～TD13、TD21、TD31～TD32がリンクポイントにより更にリンクされている。そして、これらの転送ディスクリプタには、エンドポイント1、2、3との間で転送するバケットデータに必要な情報が含まれている。

【0052】

図3（A）のリスト構造のディスクリプタは、図3（B）のCPU610（広義には処理部）上で動作するファームウェア（ホストコントローラ・ドライバ）が作成し、システムメモリ620に書き込む。即ち、ファームウェアは、システム中のエンドポイントに対してエンドポイントディスクリプタを割り当て、エンドポイント情報等に基づいてエンドポイントディスクリプタ、転送ディスクリプタを作成する。そして、これらのディスクリプタをリンクポイントでリンクさせて、システムメモリ620に書き込む。

【0053】

データ転送制御装置600（ホストコントローラ）は、システムメモリ620に書き込まれたリスト構造のディスクリプタを読み出し、エンドポイントディスクリプタや転送ディスクリプタに記述される情報に基づいて、データ転送を実行する。

【0054】

具体的には、データ転送制御装置600（ホストコントローラ）は、ED1に基づいてエンドポイント1の情報を設定し、ED1にリンクされるTD11に基づいて、エンドポイント1との間でデータ転送を行う。次に、ED2に基づいてエンドポイント2の情報を設定し、ED2にリンクされるTD21に基づいて、エンドポイント2との間でデータ転送を行う。同様にして、データ転送制御装置600は、TD31、TD12、TD32、TD13に基づいてデータ転送を実行する。

## 【0055】

以上のように、OHC I 準拠のデータ転送制御装置（ホストコントローラ）では、CPU 上で動作するファームウェア（ホストコントローラ・ドライバ）が、図3（A）に示すような複雑な構造のディスクリプタを作成しなければならない。従って、CPU の処理負荷が非常に重い。

## 【0056】

この場合、従来のUSB では、ホストの役割が割り当てられるのはパーソナルコンピュータであり、このパーソナルコンピュータは高性能のCPU を有している。従って、図3（A）に示すような複雑な構造のディスクリプタの作成も、余裕を持って行うことが可能であった。

## 【0057】

ところが、OTG のターゲットアプリケーションである小型携帯機器（デジタルカメラ、携帯電話等）に組み込まれるCPU（embedded CPU）は、パーソナルコンピュータのCPU に比べて、性能が格段に低いのが一般的である。従って、携帯機器にOTG のホスト動作を行わせると、携帯機器に組み込まれるCPU に過大な負荷がかかり、他の処理に支障が生じたり、データ転送のパフォーマンスが低下するなどの問題が生じる。

## 【0058】

## 3. 構成例

図4に、以上のような問題を解決できる本実施形態のデータ転送制御装置の構成例を示す。なお、本実施形態のデータ転送制御装置は、図4の全ての回路ブロックを含む必要はなく、その一部の回路ブロックを省略してもよい。

## 【0059】

トランシーバ10（以下、適宜Xcvrと呼ぶ）は、差動データ信号DP、DMを用いてUSB（広義にはバス）のデータを送受信する回路であり、USBの物理層（PHY）回路12を含む。より具体的にはトランシーバ10は、DP、DMのラインステート（J、K、SE0等）の生成、シリアル／パラレル変換、パラレル／シリアル変換、ビットスタフフィング、ビットアンスタフフィング、NRZIデコード、NRZIエンコードなどを行う。なお、トランシーバ10をデータ転送制御装置の外部に設けるようにしてもよい。

## 【0060】

OTGコントローラ20（広義にはステートコントローラ。以下、適宜OTGCと呼ぶ）は、OTGのSRP機能やHNP機能（図2（A）、（B）参照）を実現するための種々の処理を行う。即ち、OTGコントローラ20は、ホストの役割として動作するホスト動作のステートや、ペリフェラルの役割として動作するペリフェラル動作のステートなどを含む複数のステートの制御を行う。

## 【0061】

より具体的には、OTG規格には、デュアルロール・デバイスのAデバイス時（図1（B）、（C）参照）のステート遷移やBデバイス時のステート遷移が定義されている。OTGコントローラ20は、これらのステート遷移を実現するためのステートマシーンを含む。また、OTGコントローラ20は、USBのデータラインステートや、VBUSレベルや、IDピンのステートを検出（監視）する回路を含む。そして、OTGコントローラ20が含むステートマシーンは、これらの検出情報に基づいて、そのステート（例えば、ホスト、ペリフェラル、サスペンド又はアイドルなどのステート）を変化させる。この場合のステートの遷移は、ハードウェア回路により実現してもよいし、ファームウェアがステートコマンドをレジスタに設定することで実現してもよい。そして、ステートが遷移すると、OTGコントローラ20は、遷移後のステートに基づいて、VBUSを制御したり、DP、DMのプルアップ抵抗／プルダウン抵抗の接続／非接続を制御する。また、ホストコントローラ50（以下、適宜HCと呼ぶ）、ペリフェラルコントローラ60（以下、適宜PCと呼ぶ）のイネーブル／ディスエーブルを制御する。

## 【0062】

HC／PC切り替え回路30（HC／PC・コモン回路）は、トランシーバ10と、ホス

トコントローラ50又はペリフェラルコントローラ60との間の接続の切り替え制御を行う。また、USBのデータ(DP、DM)のラインステートの生成の指示をトランシーバ10に対して行う。なお、接続の切り替え制御は、HC/PCセクタ32により実現され、ラインステートの生成指示は、ラインステートコントローラ34により実現される。

【0063】

例えばOTGコントローラ20が、ホスト動作時(ホストステート時)にHCイネーブル信号をアクティブにすると、HC/PC切り替え回路30(HC/PCセクタ32)は、トランシーバ10とホストコントローラ50を接続する。一方、OTGコントローラ20が、ペリフェラル動作時(ペリフェラルステート時)にPCイネーブル信号をアクティブにすると、HC/PC切り替え回路30は、トランシーバ10とペリフェラルコントローラ60を接続する。このようにすることで、ホストコントローラ50とペリフェラルコントローラ60とを排他的に動作させることが可能になる。

【0064】

転送コントローラ40は、USB(広義にはバス)を介したデータ転送を制御する回路であり、ホストコントローラ50(HC)とペリフェラルコントローラ60(PC)を含む。

【0065】

ここでホストコントローラ50は、ホスト動作時(OTGコントローラ20からのHCイネーブル信号のアクティブ時)に、ホストの役割としてのデータ転送制御を行う回路である。

【0066】

即ち、ホストコントローラ50は、ホスト動作時に、HC/PC切り替え回路30によりトランシーバ10に接続される。そしてホストコントローラ50は、レジスタ部70の転送条件レジスタ部72に設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生する。そして、パケットバッファ100に確保(allocate)されたパイプ領域(PIPE0~PIPEe。以下、適宜PIPEと呼ぶ)と、そのパイプ領域に対応するエンドポイントとの間で、データ(パケット)の自動転送(処理部が介在しないハードウェア回路によるデータ転送)を行う。

【0067】

より具体的にはホストコントローラ50は、複数のパイプ転送間の調停、フレームにおける時間管理、転送のスケジューリング、再送の管理などを行う。また、パイプ転送の転送条件情報(オペレーション情報)をレジスタ部70を介して管理する。また、トランザクションの管理を行ったり、パケットを生成/分解したり、サスペンド/レジャーム/リセット状態生成の指示を行う。

【0068】

一方、ペリフェラルコントローラ60は、ペリフェラル動作時(OTGコントローラ20からのPCイネーブル信号のアクティブ時)に、ペリフェラルの役割としてのデータ転送制御を行う回路である。

【0069】

即ち、ペリフェラルコントローラ60は、ペリフェラル動作時に、HC/PC切り替え回路30によりトランシーバ10に接続される。そして、レジスタ部70の転送条件レジスタ部72に設定された転送条件情報に基づいて、パケットバッファ100に確保されたエンドポイント領域(EP0~EPe。以下適宜EPと呼ぶ)とホストとの間でデータを転送する。

【0070】

より具体的には、ペリフェラルコントローラ60は、エンドポイント転送の転送条件情報(オペレーション情報)をレジスタ部70を介して管理する。また、トランザクションの管理を行ったり、パケットを生成/分解したり、リモート・ウェイクアップ信号生成の指示を行う。

【0071】

なお、エンドポイントは、ユニークなアドレスを割り当てることができる、ペリフェラル（デバイス）上のポイント（部分）である。ホストとペリフェラル（デバイス）との間でのデータ転送は、全て、このエンドポイントを経由して行われる。また、トランザクションは、トークンパケットと、オプションなデータパケットと、オプションなハンドシェークパケットにより構成される。

【0072】

レジスタ部70は、データ転送（パイプ転送、エンドポイント転送）制御、バッファアクセス制御、バッファ管理、割り込み制御、ブロック制御、或いはDMA制御などを行うための種々のレジスタを含む。なお、レジスタ部70が含むレジスタは、RAMなどのメモリにより実現してもよいし、Dフリップフロップなどにより実現してもよい。また、レジスタ部70のレジスタは、1つにまとめずに、各ブロック（HC、PC、OTGC、Xcvr等）に分散して配置してもよい。

【0073】

レジスタ部70は、転送条件レジスタ部72を含む。そして、この転送条件レジスタ部72は、ホスト動作時にパケットバッファ100に確保されるパイプ領域（PIPE0～PIPEe）とエンドポイントとの間でのデータ転送の転送条件情報（転送制御情報）を記憶するレジスタを含む。これらの各転送条件レジスタは、パケットバッファ100の各パイプ領域に対応して設けられる。

【0074】

なお、ペリフェラル動作時には、パケットバッファ100にはエンドポイント領域（EP0～EPe）が確保される。そして、転送条件レジスタ部72に設定された転送条件情報に基づいて、データ転送制御装置とホストとの間でのデータ転送が行われる。

【0075】

バッファコントローラ80（FIFOマネージャ）は、パケットバッファ100に対するアクセス（リード／ライト）制御や領域管理を行う。より具体的には、CPU（広義には処理部）、DMA（Direct Memory Access）、USBによるパケットバッファ100へのアクセス・アドレスを生成・管理する。また、CPU、DMA、USBによるパケットバッファ100へのアクセスの調停を行う。

【0076】

例えば、ホスト動作時には、バッファコントローラ80は、インターフェース回路110（CPU又はDMA）とパケットバッファ100の間のデータ転送経路と、パケットバッファ100とホストコントローラ50（USB）の間のデータ転送経路を設定（確立）する。

【0077】

一方、ペリフェラル動作時には、バッファコントローラ80は、インターフェース回路110（CPU又はDMA）とパケットバッファ100の間のデータ転送経路と、パケットバッファ100とペリフェラルコントローラ60（USB）の間のデータ転送経路を設定する。

【0078】

パケットバッファ100（FIFO、パケットメモリ、バッファ）は、USBを介して転送されるデータ（送信データ又は受信データ）を一時的に記憶（バッファリング）するものである。このパケットバッファ100は、例えばRAM（Random Access Memory）などにより構成できる。なお、パケットバッファ100をデータ転送制御装置の外部に設けてもよい（外付けメモリにしてもよい）。

【0079】

ホスト動作時には、パケットバッファ100はパイプ転送用のFIFO（First-In-First-Out）として使用される。即ち、パケットバッファ100には、USB（バス）の各エンドポイントに対応するように、パイプ領域PIPE0～PIPEe（広義にはバッファ領域）が確保される。また、各パイプ領域PIPE0～PIPEeには、それに対応する各エンドポイントとの間で転送されるデータ（送信データ又は受信デー

タ)が記憶される。

【0080】

一方、ペリフェラル動作時には、パケットバッファ100はエンドポイント転送用のFIFOとして使用される。即ち、パケットバッファ100には、エンドポイント領域EP0～EPe(広義にはバッファ領域)が確保される。また、各エンドポイント領域EP0～EPeには、ホストとの間で転送されるデータ(送信データ又は受信データ)が記憶される。

【0081】

なお、パケットバッファ100に確保されるバッファ領域(ホスト動作時にパイプ領域に設定され、ペリフェラル動作時にエンドポイント領域に設定される領域)は、先に入力された情報が先に出力されるような記憶領域(FIFO領域)に設定されている。

【0082】

また、PIPE0は、コントロール転送用のエンドポイント0に専用のパイプ領域であり、PIPEa～PIPEeは、任意のエンドポイントに割り当て可能な汎用のパイプ領域である。

【0083】

即ち、USBでは、エンドポイント0がコントロール転送に専用のエンドポイントに設定される。従って、本実施形態のようにPIPE0をコントロール転送に専用のパイプ領域にすることで、ユーザが混乱するのを防止できる。また、PIPEa～PIPEeを、任意のエンドポイントに割り当て可能なパイプ領域にすることで、エンドポイントに対応するパイプ領域を動的に変化させることが可能になる。これにより、パイプ転送のスケジューリングの自由度を向上でき、データ転送の効率化を図れる。

【0084】

なお本実施形態では、バッファ領域(パイプ領域又はエンドポイント領域)は、マックスパケットサイズMaxPktSize(広義にはページサイズ)とページ数BufferPageにより、その領域サイズRSizeが設定される( $RSize = MaxPktSize \times BufferPage$ )。このようにすることで、バッファ領域の領域サイズや面数(ページ数)を任意に設定できるようになり、パケットバッファ100のリソースの有効利用を図れる。

【0085】

インターフェース回路110は、USBとは異なる他のバスであるDMA(システムメモリ)バスやCPUバスと、パケットバッファ100との間でのデータ転送を行うための回路である。このインターフェース回路110は、パケットバッファ100と外部のシステムメモリとの間で、DMA転送を行うためのDMAハンドラ回路112を含む。また、パケットバッファ100と外部のCPUとの間で、PIO(Parallel I/O)転送を行うためのCPUインターフェース回路114を含む。なお、CPU(処理部)をデータ転送制御装置に内蔵させてもよい。

【0086】

クロックコントローラ120は、内蔵PLL又は外部入力クロックに基づいて、データ転送制御装置の内部で使用する各種のクロックを生成する。

【0087】

#### 4. パイプ領域

本実施形態では図5(A)に示すように、ホスト動作時に、パケットバッファ100にパイプ領域PIPE0～PIPEeが確保(allocate)される。そして、この各パイプ領域PIPE0～PIPEeとペリフェラルの各エンドポイントの間で、データが転送される。

【0088】

ここで、本実施形態のパイプ領域の「パイプ」は、USBで定義される「パイプ」(デバイス上のエンドポイントとホスト上のソフトウェアとの間の関連を表す論理的な抽象化、論理的な経路)とは若干意味合いが異なる。

**【0089】**

本実施形態のパイプ領域は、図5(A)に示すように、USB(バス)に接続されるペリフェラルが有する各エンドポイントに対応して、パケットバッファ100上に確保される。例えば図5(A)において、パイプ領域PIPE aはペリフェラル1のエンドポイント1(バルクIN)に対応し、PIPE bはペリフェラル1のエンドポイント2(バルクOUT)に対応する。また、PIPE cはペリフェラル2のエンドポイント1(バルクIN)に対応し、PIPE dはペリフェラル2のエンドポイント2(バルクOUT)に対応する。また、PIPE eはペリフェラル3のエンドポイント1(インタラプトIN)に対応する。なお、PIPE 0は、コントロール転送のエンドポイント0に専用のパイプ領域である。

**【0090】**

そして図5(A)の例では、パイプ領域PIPE aとペリフェラル1のエンドポイント1との間でUSBのバルクIN転送が行われ、PIPE bとペリフェラル1のエンドポイント2との間ではバルクOUT転送が行われる。また、PIPE cとペリフェラル2のエンドポイント1との間ではバルクIN転送が行われ、PIPE dとペリフェラル2のエンドポイント2との間ではバルクOUT転送が行われる。また、PIPE eとペリフェラル3のエンドポイント1との間ではインタラプトIN転送が行われる。

**【0091】**

このように本実施形態では、パイプ領域(汎用)とそれに対応するエンドポイントとの間では、任意のデータ転送(アイソクロナス転送、バルク転送、インタラプト転送)を行うことができる。

**【0092】**

そして本実施形態では、パイプ領域とそれに対応するエンドポイントの間では、所与のデータ単位(トータルサイズで指定されるデータ単位)のデータが転送される。この場合のデータ単位としては、例えばIRP(I/Oリクエストパケット)により転送要求されるデータ単位又はそれを適当なサイズに分割したデータ単位などを考えることができる。エンドポイントに対するこのデータ単位のデータ転送(一連のトランザクション)を、本実施形態における「パイプ」と呼ぶことができる。そして、そのような「パイプ」のデータ(送信データ、受信データ)を記憶する領域がパイプ領域になる。

**【0093】**

パイプ領域を用いた所与のデータ単位の転送が終了すると、そのパイプ領域は解放することができる。そして、解放されたパイプ領域は、任意のエンドポイントに割り当てることができる。このように本実施形態では、パイプ領域とエンドポイントとの対応づけを、動的に変化させることができる。

**【0094】**

また本実施形態では図5(B)に示すように、ペリフェラル動作時には、パケットバッファ100にエンドポイント領域EP0~EPeが確保(設定)される。そして、この各エンドポイント領域EP0~EPeとホスト(ホストコントローラ、システムメモリ)との間で、データが転送される。

**【0095】**

このように本実施形態では、パケットバッファ100のバッファ領域を、ホスト動作時にはパイプ領域に割り当て、ペリフェラル動作時にはエンドポイント領域に割り当てている。これにより、ホスト動作時とペリフェラル動作時とでパケットバッファ100のリソースを共用(兼用)することが可能になり、パケットバッファ100の使用記憶容量を節約できる。

**【0096】**

なおパイプ領域、エンドポイント領域の個数は6個に限定されず任意である。

**【0097】**

5. ホストコントローラによるショートパケット自動転送

さて、ホストが、IRPデータ(所与のデータ単位)をパケットに分割してエンドポイン



トに転送する場合に、図6（A）のA1に示すように、最後のパケットがショートパケット（マックスパケットサイズ未満のサイズのパケット）になる場合がある。そして、図3（A）に示すようなリスト構造のディスクリプタでトランザクションを管理するOHC Iでは、このディスクリプタ（TD）の中に、最後のパケットがショートパケットを否かを示すショートパケット情報が含まれている。従って、データ転送制御装置（ホストコントローラ）は、ディスクリプタに含まれるショートパケット情報を参照することで、最後のパケットがショートパケットか否かを判断できる。

【0098】

しかしながら、前述のように、図3（A）に示すリスト構造のディスクリプタを作成する処理は、ファームウェア（処理部）にとって負荷の重い処理になる。このため、本実施形態では、図4に示すような構成を採用し、ホストコントローラ50が、パイプ領域とエンドポイントの間で、データを自動転送するようにしている。

【0099】

ところが、OHC Iのディスクリプタを使用せずに図4の構成でデータを自動転送すると、OHC Iのディスクリプタに含まれるショートパケット情報を利用できない。従って、最後のパケットがショートパケットか否かをファームウェアがトランザクション毎にチェックしなければならなくなり、図4の構成でデータの自動転送を実現した意味が薄れてしまう。即ち図4の構成では、ファームウェアは、自動転送開始を指示した後、割り込みが発生するまでは、データ転送の制御に関与しなく済むという利点がある。しかしながら、ショートパケットのチェック処理をファームウェアがトランザクション毎に行わなければならないとなると、このような利点が失われてしまう。

【0100】

またUSBのOUTトランザクションにおいては次のような課題があることも判明した。即ち、図6（B）のA2に示すようにIRPデータの最終パケットのペイロードサイズがマックスパケットサイズになる場合に、A3に示すIRPデータの終端を認識できないペリフェラルが存在することが判明した。

【0101】

従って、このようなペリフェラルがUSBに接続されている場合には、ホストは故意に、データ長（ペイロードサイズ）が零のショートパケット（零長パケット）を転送して、そのペリフェラルにIRPデータの終端を認識させることが望ましい。

【0102】

しかしながら、本実施形態のホストコントローラ50は、トランザクション単位でデータ転送の制御を行っている。このため、何ら工夫を施さないと、ホストコントローラ50がIRPデータの終端を判断できない。従って、ファームウェアがIRPデータの終端を判断しなければならなくなり、ファームウェアの処理負荷が増加してしまうという課題がある。

【0103】

そこで本実施形態では、ホストコントローラ50（広義には転送コントローラ、データ転送制御装置。以下の説明でも同様）がエンドポイントに対してデータ（パケット）を自動転送する際に、ショートパケット（零長パケットを含む）を自動検出して転送する手法を採用している。

【0104】

より具体的には本実施形態では、図4のレジスタ部70のレジスタに、マックスパケットサイズと転送データのトータルサイズを設定する。このレジスタは、各パイプ領域毎に設けられる転送条件レジスタであり、マックスパケットサイズとトータルサイズは、転送条件情報としてファームウェア（処理部）により設定される。

【0105】

そしてホストコントローラ50は、エンドポイントに対するトランザクション（トークンパケット、オプションなデータパケット、オプションなハンドシェイクパケットにより構成されるエンドポイントに対するサービス）を自動発生し、転送データの残りデータ

サイズを演算しながら、パイプ領域からエンドポイントにデータを自動転送する。

【0106】

即ち、ホストコントローラ50は、レジスタに設定されたトータルサイズ（各パイプ領域の転送データのトータルサイズ）を、トランザクションが完了する毎に、転送されたデータのサイズ分だけ減算する。そして、減算により得られた残りデータサイズ（各パイプ領域の転送データの残りデータサイズ）を、トータルサイズのレジスタに書き戻す。

【0107】

そして、今回のトランザクションでの残りデータサイズ（RemainSize）が、マックスパケットサイズ（MaxPktSize）未満の場合には、次のトランザクションを自動発生して、パイプ領域（パケットバッファ100）の残り転送データを、ショートパケットとして、そのパイプ領域に対応するエンドポイントに自動転送する。

【0108】

また本実施形態では、今回のトランザクションで転送するパケットのデータサイズがマックスパケットサイズであり、且つ、転送データの残りデータサイズが零の場合には、次のトランザクションを自動発生して、データ長が零（ペイロードサイズが零）のショートパケットをエンドポイントに自動転送する。これにより、ペリフェラルは、IRPデータの終端を認識できるようになる。

【0109】

以上のようにすることで、本実施形態では、図3（A）のOHC Iのディスクリプタのショートパケット情報を利用せずにショートパケットを検出して転送できるようになる。従って、ファームウェアがOHC Iのディスクリプタを作成する必要がなくなる。また、ファームウェアが、残りデータサイズを監視してショートパケットを検出したり、IRPデータの終端を検出したりする必要もなくなる。このため、ファームウェアの処理負荷を格段に軽減でき、処理を効率化できる。

【0110】

次に、ホストコントローラ50によるショートパケット、零長パケットの転送手法について図7、図8を用いて説明する。なお、これらの図において、「H→P」は「ホストからペリフェラルにパケットが転送されること」を表し、「P→H」は「ペリフェラルからホストにパケットが転送されること」を表す。

【0111】

図7は、OUTトランザクションにおいて、図6（A）のようにIRPデータの転送がショートパケットで完結する場合の信号波形例である。

【0112】

ファームウェアが自動トランザクションの開始指示を行うと、D1に示すようにPipeXTranGo（PIPE0～PIPEeのいずれかであるPIPExについてのファームウェアから転送要求信号）がアクティブになり、D2に示すようにPipeTranGo（ホストコントローラ50内のHCシーケンス管理回路からの転送要求信号）がアクティブになる。すると、ホストコントローラ50は、D3に示すようにOUTトークンパケットをペリフェラルに転送し、D4に示すようにデータパケット（マックスパケットサイズ）を転送する。そして、D5に示すようにペリフェラルからハンドシェイクパケット（ACK）が返ってくると、D6に示すようにTranCmpACK（HCシーケンス管理回路への転送正常終了の通知信号）がアクティブになる。

【0113】

この場合に図7では、D7に示すようにRemainSize（残りデータサイズ＝トータルサイズ－転送済みサイズ）がMaxPktSize未満になっている。すると、ホストコントローラ50は、IRPの最後ではなく転送すべきデータがまだ有るため（PipeXTranGoがアクティブであるため）、D9に示すようにPipeTranGoをアクティブにして、次のトランザクションを自動発生する。

【0114】

そして、ホストコントローラ50は、D10に示すようにOUTトークンパケットをペリ

フェラルに転送し、D11に示すようにショートパケットであるデータパケットをペリフェラルに転送する。そして、D12に示すようにペリフェラルからハンドシェークパケット(ACK)が返ってくると、D13に示すようにPipeXTranComp(ファームウェアへのIRPデータ単位の転送終了割り込み信号)がアクティブになる。この割り込みにより、ファームウェアは、そのパイプ(IRPデータ)についての全ての転送が完了したことを知ることができる。

【0115】

なお、PipeXTranCompがアクティブになると、D14に示すようにPipeXTranGoが非アクティブになり、そのパイプが非転送状態であることが示されるようになる。

【0116】

図7に示すように本実施形態では、一連のマックスパケットサイズのパケットの自動転送のみならず、ショートパケットの自動転送も行われる。このため、ファームウェアは、D1に示すようにPipeTranGoをアクティブにした後、D13に示すように割り込みが発生するまでの期間を、他の処理に使用できるようになる。これにより、ファームウェアの処理を効率化できる。

【0117】

例えば、一連のマックスパケットサイズのパケットの自動転送だけを行い、ショートパケットの自動転送を行わない手法であると、ファームウェアが、最後のショートパケットを検出して転送指示する必要があるため、パケットの自動転送を行う意味が薄れてしまう。ショートパケットの自動転送を行う本実施形態の手法によれば、このような事態を防止できる。

【0118】

図8は、OUTトランザクションにおいて、図6(B)のようにIRPデータの転送がマックスパケットサイズのパケットで完結する場合の信号波形例である。

【0119】

図8では、図7と異なり、E7に示すようにRemainSize(残りデータサイズ=トータルサイズ-転送済みサイズ)が零であると共に、E8に示すようにCrtTranSize(今回に転送するデータサイズ)がマックスパケットサイズ(MaxPktSize)になっている。この場合には、ホストコントローラ50は、零長のパケットを転送するために、E9に示すようにPipeTranGoをアクティブにして、次のトランザクションを自動発生する。

【0120】

そして、ホストコントローラ50は、E10に示すようにOUTトークンパケットをペリフェラルに転送し、E11に示すように零長のデータパケット(空パケット)をペリフェラルに転送する。そして、E12に示すようにペリフェラルからハンドシェークパケット(ACK)が返ってくると、E13に示すようにPipeXTranCompがアクティブになる。この割り込みにより、ファームウェアは、そのパイプについての全ての転送が完了したことを知ることができる。なお、PipeXTranCompがアクティブになると、E14に示すようにPipeXTranGoが非アクティブになる。

【0121】

図8に示すように本実施形態では、一連のマックスパケットサイズのパケットの自動転送のみならず、零長のパケットの自動転送も行われる。従って、図6(B)のA3に示すIRPデータの終端を認識できないペリフェラルが存在する場合にも、零長のパケットをそのペリフェラルに転送することで、IRPデータの終端を認識させることができる。これにより、適正なデータ転送を実現できる。

【0122】

また本実施形態によれば、ファームウェアが、転送中のIRPデータのサイズを管理しなくても、IRPデータの終端での零長のパケットは自動生成される。即ち、ファームウェアは、E1に示すようにPipeTranGoをアクティブにした後、E13に示すよう

に割り込みが発生するまでの期間を、他の処理に使用できるようになる。従って、ファームウェア（ソフトウェア）の処理負荷を軽減できると共に、ソースコード開発の手間も軽減できる。

#### 【0123】

なお、本実施形態では、零長パケットの自動転送のオン、オフを、バッファ領域（パイプ領域）毎に設定できるようになっている。即ち、バッファ領域毎に設けられた転送条件レジスタに、零長パケットの自動転送指示（AutoZeroLen）を設定できる。

#### 【0124】

図9に、ホストコントローラ50によるショートパケット、零長パケットの自動転送処理のフローチャートを示す。

#### 【0125】

まず、トータルサイズのレジスタの値が、MaxPktSize（マックスパケットサイズ）未満か否かを判断する（ステップS101）。このトータルサイズレジスタは、レジスタ部70に含まれる転送条件レジスタである。そして、このトータルサイズレジスタには、転送データのトータルサイズ（IRPのサイズ）が初期値としてファームウェアにより設定される。

#### 【0126】

トータルサイズレジスタの値がMaxPktSize未満ではない場合には、CrtTranSize（今回のトランザクションで転送されるパケットのサイズ）にMaxPktSizeを設定する（ステップS102）。そして、パイプ領域のデータがCrtTranSize未満か否かを判断し（ステップS103）、CrtTranSize未満の場合にはパイプ領域（FIFO）にデータが蓄積されるのを待つ（ステップS104）。一方、CrtTranSize未満でなくなった場合には、パイプ領域のデータ（パケット）を送信する（ステップS105）。そして、ペリフェラルからACKが返ってくると、トータルサイズレジスタにRemainSize（残りデータサイズ＝トータルサイズ－転送済みサイズ）を設定する（ステップS106、S107）。

#### 【0127】

次に、RemainSizeが零か否かを判断し（ステップS108）、零ではない場合（残りデータが有る場合）にはステップS101に戻る。一方、RemainSizeが零の場合（転送データの末尾の場合）には、AutoZeroLen（零長パケット自動転送）の設定が有効か否かを判断する（ステップS109）。そして、AutoZeroLenの設定が有効である場合には、図8のE11に示すように零長パケットを送信する（ステップS110）。そして、ペリフェラルからACKが返ってくるのを待つ（ステップS111）。

#### 【0128】

ステップS101で、トータルサイズレジスタの値がMaxPktSize未満と判断された場合には、CrtTranSizeにトータルサイズレジスタの値を設定する（ステップS112）。そして、パイプ領域のデータがCrtTranSize未満か否かを判断し（ステップS113）、CrtTranSize未満の場合にはパイプ領域（FIFO）にデータが蓄積されるのを待つ（ステップS114）。一方、CrtTranSize未満でなくなった場合には、パイプ領域のデータを図7のD11に示すようにショートパケットとして送信する（ステップS115）。そして、ペリフェラルからACKが返ってくるのを待つ（ステップS116）。

#### 【0129】

図10に、零長パケットの自動転送を実現する回路の構成例を示す。この図10の転送指示回路210はホストコントローラ50（ターゲットパイプ管理回路）内に設けられる。

#### 【0130】

転送指示回路210は、残りデータサイズ演算回路212、零長パケット転送判断回路214、レジスタ216（CrtTranSize）、レジスタ218（AutoZeroFlg）を含む。

## 【0131】

転送指示回路210は、レジスタ216のCrtTranSizeで指定されるデータサイズの packets を転送するように、転送指示を行う。このレジスタ216(CrtTranSize)には、レジスタ200からのTotalSizeがMaxPktSize未満の場合には、TotalSizeが設定され、TotalSizeがMaxPktSizeより大きい場合には、MaxPktSizeが設定される。

## 【0132】

残りデータサイズ演算回路212は、レジスタ200、202(転送条件レジスタ)からのTotalSize、MaxPktSizeを受ける。そして、これらのTotalSize、MaxPktSizeに基づいて、RemainSizeを計算する。そして、得られたRemainSizeをレジスタ200に書き戻す。なお、レジスタ200には、初期値として、転送データ(IRPデータ)のトータルサイズが設定されている。

## 【0133】

図6(B)のA2に示すようにIRPデータの最終パケットのペイロードサイズがマックスパケットサイズである場合には、この最終パケットの転送時には、CrtTranSize=MaxPktSizeになると共にRemainSize=0になる。この時に、零長パケット転送判断回路214は、レジスタ218のフラグであるAutoZeroFlgを立てる。そして、転送指示回路210は、CrtTranSize(=MaxPktSize)で指定されるデータサイズの packets を転送を指示する。これにより、図8のE4に示すように、MaxPktSizeの packets が転送される。そして、パケット転送が完了して図8のE6に示すようにTranCmpAckがアクティブになっても、PipeXtranGoはアクティブのままとなるため、E9に示すようにPipeTranGoが再びアクティブになる。

## 【0134】

この時に、転送指示回路210は、レジスタ218のフラグAutoZeroFlgが立っているため、レジスタ216のCrtTranSizeを強制的に零に設定し、パケット転送を指示する。これにより、図8のE11に示すように零長の packets が転送される。そして、転送指示回路210は、転送が正常終了すると、レジスタ218のフラグAutoZeroFlgをクリアする。

## 【0135】

以上のようにすることで、転送相手が図6(B)のA3に示すIRPデータの終端を認識できないような場合にも、零長の packets を転送することで、これを認識させることが可能になる。

## 【0136】

6. ペリフェラルコントローラによる零長パケット自動転送

さて、本実施形態のデータ転送制御装置は、図5(A)、(B)で説明したように、OTGのデュアルロール・デバイス機能により、ホストとしての動作のみならず、ペリフェラルとしても動作することができる。そして、このペリフェラル動作時には、図4のペリフェラルコントローラ60がデータ転送を制御する。

## 【0137】

そして、このように本実施形態のデータ転送制御装置がペリフェラルとして動作する場合に、次のような問題が起こることが判明した。即ち、USBのINTランザクションにおいて、図6(B)のA2に示すようにIRPデータの最終パケットのペイロードサイズがマックスパケットサイズになる場合に、A3に示すIRPデータの終端を認識できないホストが存在することが判明した。

## 【0138】

そして、このようなホストが存在すると、ペリフェラルからホストに送信すべきデータ(DMA転送データ)が無くなった場合にも、ホストがINTトークンを送信してくる。従って、ペリフェラルは、零長パケットをホストに送信して、ホストにIRPデータの終端を認識させることが望ましい。

## 【0139】

この場合に、例えば、システムメモリからパケットバッファ100へのDMA転送完了後、パケットを送信する毎に、ファームウェアが、ホストからのINTトークンに対して零長パケットを送信するか否かを判断するという手法も考えることができる。

## 【0140】

しかしながら、この手法によるとファームウェアの処理負荷が過大になり、データ転送を効率化できない。

## 【0141】

そこで本実施形態では、ペリフェラルコントローラ60（広義には転送コントローラ、データ転送制御装置。以下の説明でも同様）が、DMA転送完了後にホストからINTトークンを受信した場合に、零長パケットを自動転送する手法を採用している。

## 【0142】

より具体的には本実施形態では、図4のDMAハンドラ回路112（広義にはインターフェース回路）が、システムメモリのバス（広義には他のバス）とパケットバッファ100との間でDMA（Direct Memory Access）転送を行う。

## 【0143】

そしてペリフェラルコントローラ60は、DMA転送が完了しており、転送すべきデータが無い場合（パケットバッファ100のエンドポイント領域の残りデータが零の場合）には、ホストからのINTトークンに対して、零長のパケットをホストに自動転送する。

## 【0144】

このようにすることで、本実施形態では、ホストがIRPデータの終端を認識できない場合にも、そのホストに対して零長のパケットを送信することで、IRPデータの終端を認識させることが可能になる。これにより、適正なデータ転送を実現できる。

## 【0145】

しかも、本実施形態では、このような零長パケットの自動転送を、ファームウェアの処理が介在することなく実現される。従って、ファームウェアの処理の効率化を図れる。

## 【0146】

次に、零長パケットの転送手法について図11を用いて説明する。

## 【0147】

図11は、INTランザクションにおいて、図6（B）のようにIRPデータの転送がマックスパケットサイズのパケットで完結する場合の信号波形例である。

## 【0148】

ホストがF1に示すようにINTトークンパケットをペリフェラルコントローラ60に転送すると、ペリフェラルコントローラ60は、F2に示すようにデータパケット（マックスパケットサイズ）をホストに転送する。そして、F3に示すようにホストからハンドシェークパケット（ACK）が返ってくる。

## 【0149】

ここで、例えばF2に示すマックスパケットサイズのデータパケットが、IRPデータの最終パケットであったとする。この時に、転送相手のホストがIRPデータの終端を認識できないホストである場合には、F4に示すように、IRPデータの終端に達したにもかかわらず、ペリフェラルコントローラ60にINTトークンを転送してくる。

## 【0150】

そこで本実施形態では、このようなホストにIRPデータの終端を認識させるために、F7に示すようにEnShortPkt（ショートパケットの転送許可）をアクティブにする。即ち、F2に示すように前回のランザクションのパケットのペイロードサイズがマックスパケットサイズであり、DMA転送が完了しており、且つ、パケットバッファ100のエンドポイント領域（広義にはバッファ領域）の残りデータが零の場合に、EnShortPktをアクティブにする。これにより、F4、F5に示すように、ペリフェラルコントローラ60は、ホストからのINTトークンに対して零長のパケット（ショートパケットの1つ）を自動転送するようになる。そして、この零長のパケットを受信したホスト

は、IRPデータが終端であることを認識して、F 6に示すようにハンドシェークパケット(ACK)を転送してくる。

【0151】

このように本実施形態では零長パケットを自動転送することで、ファームウェアの処理負荷を増やすことなく、IRPデータの終端をホストに適正に認識させることに成功している。

【0152】

なお、本実施形態では、ペリフェラルコントローラ60による零長パケットの自動転送のオン、オフを、バッファ領域(エンドポイント領域)毎に設定できるようになっている。

【0153】

図12に、ペリフェラルコントローラ60による零長パケットの自動転送処理のフローチャートを示す。

【0154】

まず、DMA転送が完了したか否かを判断し、完了した場合にはDMA完了情報を保持する(ステップS121、S122)。そして、ホストからINTトークンを受信した場合には、エンドポイント領域のデータをUSBを介してホストに送信し、ACKがホストから返ってくるのを待つ(ステップS123、S124、S125)。

【0155】

次に、ACKInfo(ACK情報)がNormalACKか否かを判断する(ステップS126)。即ち本実施形態では、マックスパケットサイズのパケットをホストに送信し、それに対してホストからACKが返ってきた場合には、ACKInfoにNormalACK(マックスパケットサイズのパケットの送信が成功したことを示す情報)が設定される。一方、ショートパケットをホストに送信し、それに対してホストからACKが返ってきた場合には、ACKInfoにShortACK(ショートパケットの送信が成功したことを示す情報)が設定される。

【0156】

ACKInfo(ACK情報)がNormalACKの場合には、DMA転送が完了しているか否かを判断する(ステップS127)。この判断は、ステップS122で保持された情報に基づいて行う。

【0157】

DMA転送が完了している場合には、FIFODataRemain(バッファ領域の残りデータサイズ)が零か否かを判断する(ステップS128)。即ち、転送すべきデータが零か否かを判断する。

【0158】

FIFODataRemainが零の場合には、AutoZeroLen(零長パケット自動転送)の設定が有効か否かを判断する(ステップS129)。このAutoZeroLenは、ファームウェアによりレジスタ部70のレジスタ(転送条件レジスタ)に設定される。

【0159】

AutoZeroLenの設定が有効の場合には、EnShortPktビット(ショートパケットの転送許可)をセットする(ステップS130)。そして、ホストからINTトークンが返ってくるのを待ち、返ってきた場合には、零長パケットを送信する(ステップS131、S132)。そして、ホストからACKが返ってきたら、EnShortPktビットをクリアする(ステップS133、S134)。

【0160】

図13に、零長パケットの自動転送を実現する回路の構成例を示す。この回路はペリフェラルコントローラ60(トランザクション管理回路)内に設けられる。

【0161】

最終パケット判断回路230は、DMAcmp(DMA転送完了)と、ACKInfo(前回のトランザクションでマックスパケットサイズのパケットの送信に成功したか、ショ

ートパケットの転送に成功したかを示すACK情報)と、FIFODataRemain(バッファ領域の残りデータサイズ)を受ける。そして、マックスパケットサイズのパケットで完結したか否かを判断する。即ち、DMA転送が完了しており(DMAcmp)、前回のトランザクションのパケットがマックスパケットサイズのパケットであり(ACKInfo)、バッファ領域の残りデータサイズが零の場合に(FIFODataRemain)、ZeroPktをアクティブにする。

【0162】

ショートパケット転送許可回路240は、ZeroPkt、AutoZeroLen(零長パケット自動転送)を受ける。そして、ZeroPkt、AutoZeroLenが共にアクティブの場合には、EnShortPkt(ショートパケット転送許可)をアクティブにする。

【0163】

転送指示回路242は、EnShortPktがアクティブの状態、INTTranReq(ホストからのINTランザクション要求)を受けた場合に、FIFOReadReq(バッファ領域のデータの読み出し要求)をアクティブにする。この場合に、バッファ領域(エンドポイント領域)にはデータが無く、空であるため、結果的に、ホストからのINTトークンに対して零長パケットが転送されることになる。

【0164】

7. 転送条件レジスタ(共用レジスタ)

本実施形態では図14に示すように、ホスト動作時には、パイプ領域PIPE0~PIPEeとエンドポイントとの間で行われるデータ転送の転送条件情報(転送方向、転送種別、マックスパケットサイズ又はページ数等)が、転送条件レジスタTREG0~TREGeに設定される。即ち、PIPE0、PIPEa、PIPEb、PIPEc、PIPED、PIPEeの転送条件情報は、各々、TREG0、TREGa、TREGb、TREGc、TREGd、TREGeに設定(記憶)される。この設定は、例えばファームウェア(CPU)により行われる。

【0165】

そしてホストコントローラ50(広義には転送コントローラ)は、転送条件レジスタTREG0~TREGeに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを発生する。そして、パイプ領域とそれに対応するエンドポイントとの間で、データ(パケット)を自動転送する。

【0166】

このように本実施形態では、各パイプ領域(バッファ領域)に対応して各転送条件レジスタが設けられ、この各転送条件レジスタに設定された転送条件情報に基づいて、各パイプ領域のパイプ転送(所与のデータ単位の転送)がホストコントローラ50により自動的に行われる。従って、ファームウェア(ドライバ、ソフトウェア)は、転送条件レジスタに転送条件情報を設定した後は、データ転送が完了するまで、データ転送制御に関わらずに済むようになる。そして、所与のデータ単位のパイプ転送が完了すると割り込みが発生し、転送の完了がファームウェアに伝えられる。これにより、ファームウェア(CPU)の処理負荷を格段に低減できる。

【0167】

なお、本実施形態では図15に示すようにペリフェラル動作時には、エンドポイント領域EP0~EPeとホストとの間で行われるデータ転送の転送条件情報(転送方向、転送種別、マックスパケットサイズ又はページ数等)が、転送条件レジスタTREG0~TREGeに設定される。そしてペリフェラルコントローラ60(広義には転送コントローラ)は、転送条件レジスタTREG0~TREGeに設定された転送条件情報に基づいて、エンドポイント領域とホストとの間でのデータ転送を行う。

【0168】

このように本実施形態では、転送条件レジスタTREG0~TREGeが、ホスト動作時とペリフェラル動作時とで共用(兼用)される。これにより、レジスタ部70のリソース



を節約でき、データ転送制御装置を小規模化できる。

【0169】

図16に、レジスタ部70のレジスタ構成例を示す。なお、レジスタ部70のレジスタの一部を、各ブロック（OTGC、HC、PC、Xc v r等）内に含ませてもよい。

【0170】

図16に示すように、レジスタ部70の転送条件レジスタ（TREG0～TREGeの各々）は、ホスト動作時（HC、PIPE）とペリフェラル動作時（PC、EP）で共用されるHC/PC共用レジスタ（共用転送条件レジスタ）を含む。また、ホスト動作時のみ使用されるHC（PIPE）用レジスタ（ホスト用転送条件レジスタ）を含む。また、ペリフェラル動作時のみ使用されるPC（EP）用レジスタ（ペリフェラル用転送条件レジスタ）を含む。また、パケットバッファ（FIFO）のアクセス制御などを行うためのレジスタであり、ホスト動作時とペリフェラル動作時で共用されるアクセス制御レジスタを含む。

【0171】

例えば、デュアルロール・デバイスのホスト動作時に、ホストコントローラ50（HC）は、HC/PC共用レジスタとHC用レジスタに設定される転送条件情報に基づいて、データ（パケット）を転送する。

【0172】

一方、ペリフェラル動作時には、ペリフェラルコントローラ60（PC）は、HC/PC共用レジスタとPC用レジスタに設定される転送条件情報に基づいて、データ（パケット）を転送する。

【0173】

また、ホスト動作時、ペリフェラル動作時の両方において、バッファコントローラ80は、共用アクセス制御レジスタに基づいて、パケットバッファ100へのアクセス制御（リード/ライト・アドレスの発生、データのリード/ライト、アクセスの調停等）を行うことになる。

【0174】

図16のHC/PC共用レジスタには、データの転送方向（IN、OUT又はSETUP等）、転送種別（アイソクロナス、バルク、インタラプト、コントロールなどのトランザクションの種別）、エンドポイント番号（各USBデバイスのエンドポイントに関連づけられる番号）、マックスパケットサイズ（エンドポイントが送信又は受信可能なパケットの最大ペイロードサイズ。ページサイズ）が設定される。また、バッファ領域（パイプ領域、エンドポイント領域）のページ数（バッファ領域の面数）が設定される。また、DMA接続の有無（DMAハンドラ回路112によるDMA転送の使用の有無）を指示する情報が設定される。また、零長パケット自動転送指示（トータルサイズで指定した転送がマックスパケットサイズで終了する場合に、自動で零長パケットを転送するか否かの指示）が設定される。

【0175】

HC（PIPE）用レジスタには、インタラプト転送のトークン発行周期（インタラプト・トランザクションを起動する周期、インターバル）が設定される。また、トランザクションの連続実行回数（パイプ領域間の転送比率を設定する情報。各パイプ領域のトランザクションの連続実行回数）が設定される。また、ファンクションアドレス（エンドポイントを有するファンクションのUSBアドレス）、転送データのトータルサイズ（各パイプ領域を介して転送されるデータのトータルサイズ。IRPなどのデータ単位）が設定される。また、自動トランザクションの開始指示（ホストコントローラに対する自動トランザクション処理の開始指示）が設定される。また、自動コントロール転送モードの指示（コントロール転送のセットアップステージ、データステージ、ステータスステージのトランザクションを自動発生するモードの指示）が設定される。

【0176】

PC（EP）用レジスタには、エンドポイントイネーブル（エンドポイントのイネーブル

やディスエーブルの指示)、ハンドシェーク指定(各トランザクションで行われるハンドシェークの指定)が設定される。

【0177】

パケットバッファ(FIFO)用の共用アクセス制御レジスタには、バッファ・I/Oポート(CPUによりPIO転送を行う場合のI/Oポート)が設定される。また、バッファ・フル/エンプティ(各バッファ領域のフル、エンプティの通知)、バッファ・残りデータサイズ(各バッファ領域の残りデータサイズ)が設定される。

【0178】

レジスタ部70は、インタラプト系レジスタ、ブロック系レジスタ、DMA制御レジスタなども含む。

【0179】

インタラプト系レジスタは、割り込みのステータス(要因)をCPUに対して示すためのインタラプト・ステータスレジスタ、割り込みのイネーブル、ディスエーブル(非マスク、マスク)を設定するインタラプト・イネーブルレジスタを含む。なお、割り込みには、OTGコントローラ20系、ホストコントローラ50系、ペリフェラルコントローラ60系の割り込みがある。

【0180】

ブロック系レジスタは、ブロック間で共用されるブロック間共用レジスタや、各ブロック(Xcvr、OTGC、HC、PC)内で使用されるブロック用レジスタを含む。

【0181】

ブロック間共用レジスタには、各ブロックのリセットを指示するレジスタなどがある。ブロック用レジスタには、トランシーバ10(Xcvr)を制御するためのレジスタや、OTGコントローラ20(OTGC)のステートコマンドレジスタや、ホストコントローラ50(HC)のステートコマンドレジスタや、フレーム番号を設定するレジスタなどがある。

【0182】

以上のように本実施形態では、ホスト動作時とペリフェラル動作時で共用されるレジスタ(HC/PC共用レジスタ、共用アクセス制御レジスタ)をレジスタ部70に設けている。これにより、ホスト動作時用のレジスタとペリフェラル動作時用のレジスタを全く別個に設ける場合に比べて、レジスタ部70を小規模化できる。また、CPU上で動作するファームウェア(ドライバ)から見た共用レジスタのアクセスアドレスを、ホスト動作時とペリフェラル動作時とで同一にできる。従って、ファームウェアは、これらの共用レジスタを同一アドレスで管理できるようになり、ファームウェア処理を簡素化できる。

【0183】

また、HC用レジスタや、PC用レジスタを設けることで、ホスト動作時(PIPE)の転送やペリフェラル動作時(EP)の転送に特有の転送条件を設定できる。例えば、トークン発行周期を設定することで、ホスト動作時にインタラプト転送のトークンを所望の周期で発行することが可能になる。また、連続実行回数を設定することで、ホスト動作時にパイプ領域間の転送比率を任意に設定できる。また、トータルサイズを設定することで、ホスト動作時にパイプ領域を介して自動転送されるデータのサイズを任意に設定できる。またファームウェアは、ホスト動作時に、自動トランザクションの開始を指示したり、自動コントロール転送モードのオン/オフを指示できるようになる。

【0184】

#### 8. 自動トランザクション

図17に、ホストコントローラ50の自動トランザクション(IN、OUT)処理時におけるファームウェア処理のフローチャート例を示す。

【0185】

まず、ファームウェア(処理部、ドライバ)は、図16等で説明した転送条件レジスタに転送条件情報(パイプ情報)を設定する(ステップS1)。より具体的には、転送データのトータルサイズ、マックスパケットサイズ(MaxPktSize)、ページ数(Bu

fferPage)、転送方向(IN、OUT又はSETUP)、転送種別(アイソクロナス、バルク、コントロール、インタラプト)、エンドポイント番号、パイプ領域のトランザクションの連続実行回数(転送比率)、インタラプト転送のトークン発行周期、零長パケット自動転送(AutoZeroLen)などを、転送条件レジスタに設定する。

【0186】

次に、外部のシステムメモリとパケットバッファ100の間に転送経路を設定する(ステップS2)。即ち図4のDMAハンドラ回路112を介したDMA転送経路を設定する。

【0187】

次に、ファームウェアは、DMA転送の開始指示を行う(ステップS3)。即ち、図16のDMA制御レジスタのDMA転送開始指示ビットをアクティブにする。なお、CPUによる転送では、図16のバッファ・I/Oポートにアクセスすることで、パケットバッファ100にアクセスすることが可能になる。

【0188】

次に、ファームウェアは、自動トランザクションの開始指示を行う(ステップS4)。即ち、図16のHC用レジスタ(パイプレジスタ)の自動トランザクション開始指示ビットをアクティブにする。これにより、ホストコントローラ50による、自動トランザクション処理、パケット処理(パケットの生成、分解)、スケジューリング処理が行われる。即ち、ホストコントローラ50は、トータルサイズで指定されるデータを、マックスパケットサイズのペイロードのパケットを用いて、転送方向で指定される方向(IN、OUT)で、自動転送する。

【0189】

なお、ステップS3、S4の処理の順序は問わず、自動トランザクション開始指示の後にDMA転送の開始指示を行ってもよい。

【0190】

次に、ファームウェアは、パイプ転送の完了を知らせる割り込みが発生するのを待つ(ステップS5)。そして、割り込みが発生すると、ファームウェアは、図16のインタラプト系レジスタの割り込みステータス(要因)を調べる。そして、処理が正常完了又はエラー終了する(ステップS6)。

【0191】

このように本実施形態によれば、ファームウェアは、各パイプ領域毎に転送条件情報を設定し(ステップS1)、DMA転送開始の指示(ステップS3)と自動トランザクション開始の指示(ステップS4)を行うだけで、その後のデータ転送処理はホストコントローラ50のハードウェア回路により自動的に行われるようになる。従って、図3(A)、(B)で説明したOHC I 準拠の手法に比べて、ファームウェアの処理負荷が軽減され、低性能のCPUが組み込まれる携帯機器に最適なデータ転送制御装置を提供できる。

【0192】

9. 各ブロックの詳細な構成例

次に各ブロックの詳細な構成例について説明する。

【0193】

9.1 OTGコントローラ

図18に、OTGコントローラ20の構成例を示す。

【0194】

OTGコントローラ20は、OTGレジスタ部22を含む。このOTGレジスタ部22は、OTGコントローラ20のモニタレジスタや制御レジスタを含む。またファームウェア(CPU)により書き込まれるOTGステートコマンドをデコードする回路を含む。

【0195】

またOTGコントローラ20はOTG制御回路23を含む。そして、このOTG制御回路23は、OTGステートの管理を行うOTG管理回路24、IDピンの電圧レベルを検出するID検出回路25、VBUSの電圧レベルを検出するVBUS検出回路26を含む。また、トランシーバ10からのラインステート情報(J、K、SE0等)に基づいて、U

SBバスの状態（リセット、レジューム等）を検出するラインステート検出回路27を含む。

【0196】

またOTGコントローラ20は時間計測を行うタイマ28を含む。ラインステート検出回路27は、トランシーバ10からのラインステート情報とタイマ28からの時間計測情報に基づいて、USBバスの状態を検出する。

【0197】

OTGステートを遷移させるために検出すべき情報は、ID、VBUSの電圧レベル、DP/DMのラインステートである。本実施形態のOTGコントローラ20は、これらの情報を検出し、モニタレジスタを介してファームウェア（CPU）に伝える。

【0198】

ファームウェアは、これらの検出情報に基づいて自身のステートを遷移させると共に、次に遷移すべきステートを、OTGステートコマンドを用いてOTGコントローラ20に伝える。

【0199】

OTGコントローラ20は、OTGステートコマンドをデコードし、そのデコード結果に基づいて、VBUSのドライブ制御、プルアップ/プルダウン抵抗の接続制御等を行い、図2（A）、（B）で説明したSRPやHNPを実現する。

【0200】

このように本実施形態では、ステート毎のOTG制御はOTGコントローラ20が担当し、ファームウェアはステートの遷移管理に専念できる。この結果、全てのステート制御をファームウェアで実現する場合に比べて、ファームウェア（CPU）の処理負荷を軽減できると共に、効率的なファームウェア開発が可能になる。

【0201】

なお、OTGのステート遷移の判断を、ファームウェアが行わずに、ハードウェア回路が行うようにしてもよい。或いは、OTGコントローラ20のほとんど全ての処理（例えばVBUS制御、プルアップ/プルダウン抵抗制御、ID検出、VBUS検出、ラインステート検出以外の処理）をファームウェア（ソフトウェア）により実現してもよい。

【0202】

9. 2 ホストコントローラ、ペリフェラルコントローラ

図19（A）に、ホストコントローラ50の構成例を示す。

【0203】

ホストコントローラ50はHCシーケンス管理回路52を含む。このHCシーケンス管理回路52は、パイプ転送（パイプ領域を用いたデータ転送）の調停、時間管理、パイプ転送のスケジューリング、再送管理などを行う。

【0204】

より具体的にはHCシーケンス管理回路52は、フレーム番号のカウントや、SOF（Start-Of-Frame）パケットの送信指示を行う。また、アイソクロナス転送を各フレームの先頭で優先的に実行するための処理を行ったり、インタラプト転送をアイソクロナス転送の次に優先的に取り扱うための処理を行う。また、パイプ転送の順序にしたがって各パイプ転送を指示する処理を行う。また、トランザクションの連続実行回数を管理したり、残りフレーム時間の確認処理を行う。また、ペリフェラルから返ってきたハンドシェイクパケット（ACK、NAK）に対する処理を行う。また、トランザクション実行時のエラー処理を行う。

【0205】

ホストコントローラ50はターゲットパイプ管理回路54を含む。このターゲットパイプ管理回路54は、レジスタ部70の転送条件レジスタに設定された転送条件情報のハンドリング処理などを行う。

【0206】

より具体的にはターゲットパイプ管理回路54は、転送条件情報の選択処理や、割り込み

信号の生成処理を行う。また自動トランザクションの開始が指示された場合に、そのパイプ領域の転送データのトータルサイズをロードする。そして、残り転送データサイズのカウンタ（デクリメント）処理を行う。また、バッファコントローラ80へのデータの送受信の際にバッファ（FIFO）領域の状態を確認する処理を行う。また、トランザクション管理回路56への転送指示を行う。また、予期しないショートパケットの受信の判断処理や、マックスパケットサイズ以上のパケットの受信の判断処理を行う。また、零長パケットを自動転送するモードが設定されている場合には、最後の零長パケットの送信をトランザクション管理回路56に指示する。また、自動コントロール転送モードでのシーケンス管理を行う。

#### 【0207】

ホストコントローラ50はトランザクション管理回路56を含む。このトランザクション管理回路56は、転送パケットの種類や転送順序の管理（トランザクションのシーケンス管理）を行う。また、タイムアウトの監視処理を行う。また、トランザクション終了の通知処理を行う。

#### 【0208】

ホストコントローラ50はパケットハンドラ回路58を含む。このパケットハンドラ回路58は、パケットの生成、分解処理を行う。また、PIDのチェックやCRCのデコード、エンコードを行う。また、バッファ領域のパケットのペイロードのリード、ライト処理や、SOFパケットの送信処理を行う。また、送受信データのカウンタ処理を行う。

#### 【0209】

図19（B）にペリフェラルコントローラ60の構成例を示す。

#### 【0210】

ペリフェラルコントローラ60は、トランザクション管理回路62、パケットハンドラ回路64を含む。これらのトランザクション管理回路62、パケットハンドラ回路64は、ホストコントローラ50のトランザクション管理回路56、パケットハンドラ回路58とほぼ同様の処理を行う。

#### 【0211】

### 9. 3 バッファコントローラ

図20にバッファコントローラ80の構成例を示す。

#### 【0212】

バッファコントローラ80は領域確保（allocation）回路82を含む。この領域確保回路82は、パケットバッファ100に、バッファ領域（ホスト動作時にパイプ領域に設定され、ペリフェラル動作時にエンドポイント領域に設定される領域）を確保する回路である。

#### 【0213】

領域確保回路82は領域計算回路83を含む。この領域計算回路83は、マックスパケットサイズ（広義にはページサイズ）やページ数に基づいて、バッファ領域の領域サイズ、スタートアドレス、エンドアドレスなどを計算する回路である。

#### 【0214】

例えば図21（A）に示すバッファ領域PIPE0/EP0、PIPEa/EPa、PIPEb/EPb、PIPEc/EPcでは、マックスパケットサイズ（MaxPktSize）が、各々、32、64、64、64バイトに設定され、ページ数（BufferPage）が、各々、1、1、3、2ページに設定されている。領域計算回路83は、これらのマックスパケットサイズ、ページ数などに基づいて、バッファ領域PIPE0/EP0～PIPEc/EPcの領域サイズ、スタートアドレス、エンドアドレスを計算する。例えば図21（A）において、PIPE0/EP0、PIPEa/EPa、PIPEb/EPb、PIPEc/EPcの領域サイズは、各々、32（＝32×1）、64（＝64×1）、192（＝64×3）、128（＝64×2）バイトと計算されることになる。

#### 【0215】

ポインタ割り当て回路84は、各バッファ領域の書き込みポインタWP（WP0、WPa

、WPb、WPc)、読み出しポインタRP(RP0、RPa、RPb、RPc)を、DMA用ポインタ、CPU用ポインタ、USB用ポインタに割り当てる回路である。

【0216】

例えば図21(B)に示すように、データ送信時(DMA又はCPUからパケットバッファ100を介してUSB側にデータが転送される時)であり、且つ、DMA転送使用時には、そのバッファ領域の書き込みポインタWPはDMA(DMAアクセス)用のポインタに割り当てられ、読み出しポインタRPはUSB(USBアクセス)用のポインタに割り当てられる。また、データ送信時であり且つCPU(PIO)転送使用時には、そのバッファ領域の書き込みポインタWPはCPU(CPUアクセス)用のポインタに割り当てられ、読み出しポインタRPはUSB用のポインタに割り当てられる。

【0217】

一方、図21(C)に示すように、データ受信時(USBからパケットバッファ100を介してDMA又はCPU側にデータが転送される時)であり、且つ、DMA転送使用時には、そのバッファ領域の書き込みポインタWPはUSB用ポインタに割り当てられ、読み出しポインタRPはDMA用ポインタに割り当てられる。また、データ受信時であり且つCPU転送使用時には、そのバッファ領域の書き込みポインタWPはUSB用ポインタに割り当てられ、読み出しポインタRPはCPU用ポインタに割り当てられる。

【0218】

なお、各バッファ領域の書き込みポインタWP、読み出しポインタRPのポインタ情報(位置情報)は、レジスタ部70の各転送条件レジスタ(PIPE/Eレジスタ)に保持される。

【0219】

ポインタ管理回路86は、ポインタの更新を行いながら、パケットバッファ100にアクセスするための実アドレスを生成する回路である。

【0220】

ポインタ管理回路86は、CPU用アドレス生成回路87、DMA用アドレス生成回路88、USB用アドレス生成回路89を含む。これらの生成回路87、88、89は、各々、ポインタ割り当て回路84により割り当てられたCPU用ポインタ、DMA用ポインタ、USB用ポインタに基づいて、CPU用アドレス、DMA用アドレス、USB用アドレスを生成する。また、CPU(CPUインターフェース回路)、DMA(DMAハンドラ回路)からのアクセス毎に、或いはUSB(HC又はPC)のトランザクション終了(ACK、NAKなどのハンドシェイク送受信)毎に、ポインタを更新する処理を行う。なお、更新後のポインタの情報は、領域確保回路82を介してレジスタ部70の各転送条件レジスタに書き戻される。

【0221】

バッファ管理回路90は、パケットバッファ100へのアクセスを管理する回路である。

【0222】

バッファ管理回路90はバッファインターフェース回路92を含む。このバッファインターフェース回路92は、ポインタ管理回路86からのCPU用アドレス、DMA用アドレス、USB用アドレスなどを受け、パケットバッファ100へのデータの入出力や、アドレス、出力イネーブル、ライトイネーブル、リードイネーブルなどの出力を行う。

【0223】

バッファ管理回路90は調停回路93を含む。この調停回路93は、CPU(CPUインターフェース回路)、DMA(DMAハンドラ回路)、USB(ホストコントローラ又はペリフェラルコントローラ)からのアクセスを調停する回路である。この調停結果に基づいて、CPU用アドレス、DMA用アドレス、USB用アドレスのいずれかが、パケットバッファ100のアクセス・アドレスとして出力され、CPU、DMA又はUSBとパケットバッファ100との間のデータ転送経路が設定される。

【0224】

HC/PCセクタ94は、バッファ管理回路90(バッファコントローラ80)とホス

トコントローラ50(HC)又はペリフェラルコントローラ60(PC)との間の接続の切り替え制御を行う。例えばホスト動作時には、ホストコントローラ50とバッファ管理回路90を接続し、ペリフェラル動作時には、ペリフェラルコントローラ60とバッファ管理回路90を接続する。なお、この接続の切り替え制御は、OTGコントローラ20(OTGC)からのHC/PCイネーブル信号に基づいて行う。

#### 【0225】

##### 10. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

#### 【0226】

例えば図22(A)に電子機器の1つであるプリンタの内部ブロック図を示し、図23(A)にその外観図を示す。CPU510(処理部)はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するためのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM517(システムメモリ)はCPU510のワーク領域として機能する。DMAC518は、CPU510を介さずにデータ転送を行うためのDMAコントローラである。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

#### 【0227】

USBを介してパーソナルコンピュータ、デジタルカメラ、デジタルビデオカメラなどの他の機器から送られてきたシリアル印刷データ(印字データ、画像データ)は、データ転送制御装置500によりパラレル印刷データに変換される。そして、変換後のパラレル印刷データは、CPU510又はDMAC518により、印刷処理部(プリンタエンジン)512に送られる。そして、印刷処理部512においてパラレル印刷データに対して所与の処理が施され、プリントヘッドなどからなる印刷部(データの出力処理を行う装置)514により紙に印刷されて出力される。

#### 【0228】

図22(B)に電子機器の1つであるデジタルカメラの内部ブロック図を示し、図23(B)にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521(シャッターボタン、操作ボタン等)はデジタルカメラをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM527はCPU520のワーク領域として機能する。DMAC528はDMAコントローラである。

#### 【0229】

CCD、レンズなどからなる撮像部(データの取り込み処理を行う装置)522により画像が撮像され、撮像された画像のデータは画像処理部524により処理される。そして、処理後の画像データは、CPU520又はDMAC528によりデータ転送制御装置500に送られる。データ転送制御装置500は、このパラレルの画像データをシリアルデータに変換し、USBを介してプリンタ、ストレージ装置、パーソナルコンピュータなどの他の機器に送信する。

#### 【0230】

図22(C)に電子機器の1つであるCD-RWドライブ(ストレージ装置)の内部ブロック図を示し、図23(C)にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-RWをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM537はCPU530のワーク領域として機能する。DMAC538はDMAコントローラである。

#### 【0231】

レーザ、モータ、光学系などからなる読み取り&書き込み部(データの取り込み処理を行う装置又はデータの記憶処理を行うための装置)533によりCD-RW532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、CPU530又はDMAC538によりデータ転送制御装置500に送られる。データ転送制御装置500は、このパラレルのデータをシリアルデータに変換し、USBを介して他の機器に送信する。

## 【0232】

一方、USBを介して他の機器から送られてきたシリアルデータは、データ転送制御装置500によりパラレルデータに変換される。そして、このパラレルデータは、CPU530又はDMAC538により信号処理部534に送られる。そして、信号処理部534においてこのパラレルデータに対して所与の信号処理が施され、読み取り&書き込み部533によりCD-RW532に記憶される。

## 【0233】

なお、図22(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

## 【0234】

本実施形態のデータ転送制御装置を電子機器に用いれば、OTG機能を有する電子機器を実現できる。即ち、電子機器にホストとしての役割を持たせたり、デバイスとしての役割を持たせることが可能になり、これまでに存在しなかったアプリケーションを創出できる。

## 【0235】

また本実施形態のデータ転送制御装置を電子機器に用いれば、電子機器に組み込まれるCPU(処理部)の処理負荷が軽減され、安価なCPUを用いることが可能になる。また、CPUが、データ転送制御処理以外の他の処理を余裕を持って行うことが可能になり、電子機器の性能向上や低コスト化を図れる。また、CPU上で動作するファームウェアのプログラムを簡素化でき、電子機器の開発期間の短縮化を図れる。

## 【0236】

なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ(CD-ROM、DVD)、光磁気ディスクドライブ(MO)、ハードディスクドライブ、デジタルビデオカメラ、携帯電話、スキャナ、TV、VTR、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、或いはワードプロセッサなど種々のものを考えることができる。

## 【0237】

なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

## 【0238】

例えば、本発明のデータ転送制御装置の構成は、図4等で説明した構成に限定されるものではなく、種々の変形実施が可能である。

## 【0239】

また、データ転送制御装置の各ブロック(HC、PC、OTGC等)の構成も、本実施形態で説明したものに限定されず、種々の変形実施が可能である。

## 【0240】

また、ショートパケット、零長パケットを自動転送する回路も図10、図13に示す構成に限定されず、種々の変形実施が可能である。

## 【0241】

また、明細書中の記載において広義な用語(ステートコントローラ、処理部、転送コントローラ、バス、バッファ領域等)として引用された用語(OTGコントローラ、CPU・ファームウェア、ホストコントローラ・ペリフェラルコントローラ、USB、パイプ領域・エンドポイント領域等)は、明細書中の他の記載においても広義な用語に置き換えることができる。

## 【0242】

また、本発明のうち従属請求項に係る発明においては、従属先の請求項の構成要件の一部を省略する構成とすることもできる。また、本発明の1の独立請求項に係る発明の要部を、他の独立請求項に従属させることもできる。

## 【0243】



また、本実施形態ではUSBのOTG規格への適用例を説明したが、本発明が適用されるのはOTG規格に限定されない。即ち、USBのOTGのみならず、従来のUSB1.1、USB2.0や、これらの規格を発展させた規格におけるデータ転送にも本発明は適用できる。

【図面の簡単な説明】

【図1】図1(A)、(B)、(C)は、USBのOTG規格について説明するための図である。

【図2】図2(A)、(B)は、SRPやHNPの手順について説明するための図である。

【図3】図3(A)、(B)は、OHC Iのリスト構造のディスクリプタなどについて説明するための図である。

【図4】本実施形態のデータ転送制御装置の構成例を示す図である。

【図5】図5(A)、(B)は、パイプ領域、エンドポイント領域について説明するための図である。

【図6】図6(A)、(B)は、IRPデータの転送がショートパケットやマックスパケットサイズのパケットで完結する場合について説明するための図である。

【図7】OUTランザクションにおいて、IRPデータの転送がショートパケットで完結する場合の信号波形例である。

【図8】OUTランザクションにおいて、IRPデータの転送がマックスパケットサイズのパケットで完結する場合の信号波形例である。

【図9】ホストコントローラによるショートパケット、零長パケットの自動転送処理について説明するためのフローチャートである。

【図10】ショートパケット、零長パケットの自動転送処理を実現する回路の例である。

【図11】INTランザクションにおいて、IRPデータの転送がマックスパケットサイズのパケットで完結する場合の信号波形例である。

【図12】ペリフェラルコントローラによる零長パケットの自動転送処理について説明するためのフローチャートである。

【図13】零長パケットの自動転送処理を実現する回路の例である。

【図14】データ転送制御装置のホスト時の動作について説明するための図である。

【図15】データ転送制御装置のペリフェラル時の動作について説明するための図である。

【図16】レジスタ部について説明するための図である。

【図17】ファームウェアの処理例を説明するためのフローチャートである。

【図18】OTGコントローラの詳細な構成例を示す図である。

【図19】図19(A)、(B)は、ホストコントローラ、ペリフェラルコントローラの詳細な構成例を示す図である。

【図20】バッファコントローラの詳細な構成例を示す図である。

【図21】図21(A)、(B)、(C)は、領域確保手法やポインタ割り当て手法について説明するための図である。

【図22】図22(A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。

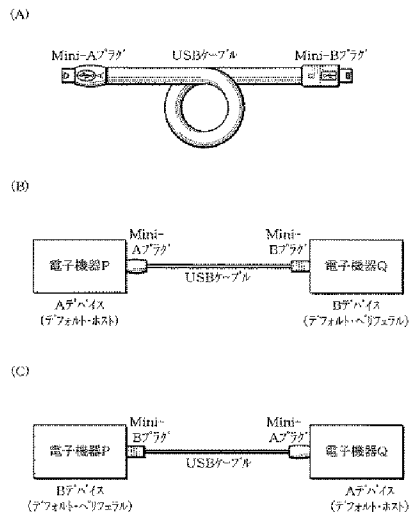
【図23】図23(A)、(B)、(C)は、種々の電子機器の外観図の例である。

【符号の説明】

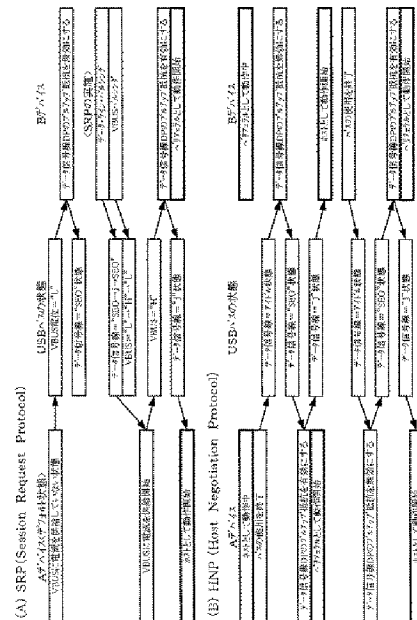
PIPE0～PIPEe   パイプ（バッファ）領域  
EP0～EPe        エンドポイント（バッファ）領域  
TREG0～TREGe   転送条件レジスタ（共用レジスタ）  
10    トランシーバ  
12    物理層回路  
20    OTGコントローラ（ステートコントローラ）  
30    HC/PC切り替え回路

- |       |                       |
|-------|-----------------------|
| 3 2   | HC／PCセレクト             |
| 3 4   | ラインステートコントローラ         |
| 4 0   | 転送コントローラ              |
| 5 0   | ホストコントローラ             |
| 6 0   | ペリフェラルコントローラ          |
| 7 0   | レジスタ部                 |
| 7 2   | 転送条件レジスタ部（共用レジスタ）     |
| 8 0   | バッファコントローラ            |
| 1 0 0 | パケットバッファ（F I F O、RAM） |
| 1 1 0 | インターフェース回路            |
| 1 1 2 | DMAハンドラ回路             |
| 1 1 4 | CPUインターフェース回路         |
| 1 2 0 | クロックコントローラ            |

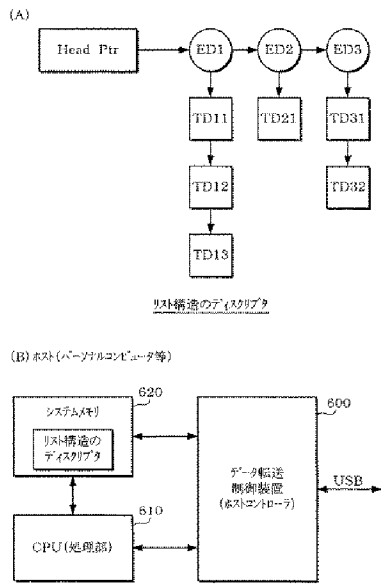
【図1】



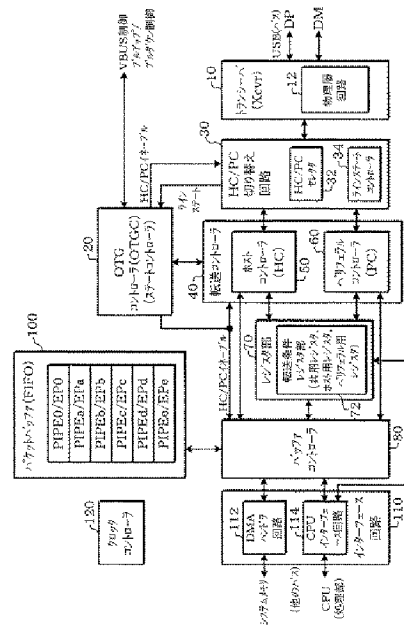
【例2】



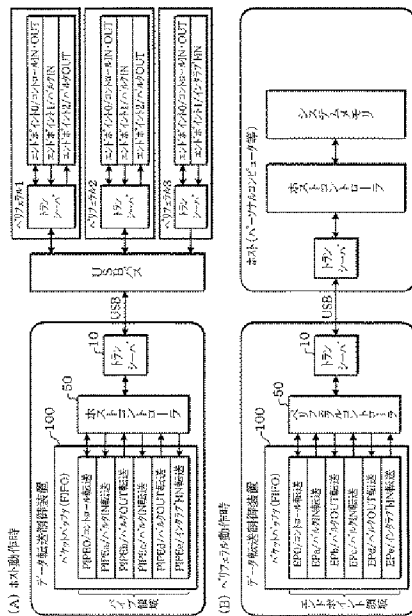
【図3】



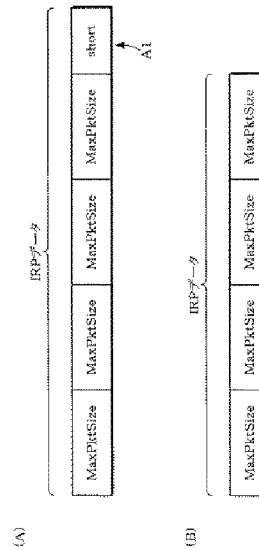
【図4】



【図5】

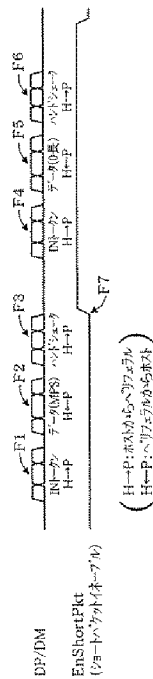


【図6】

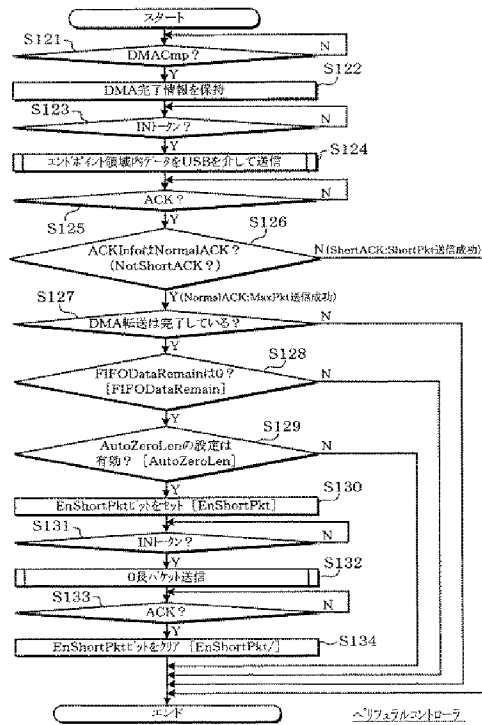




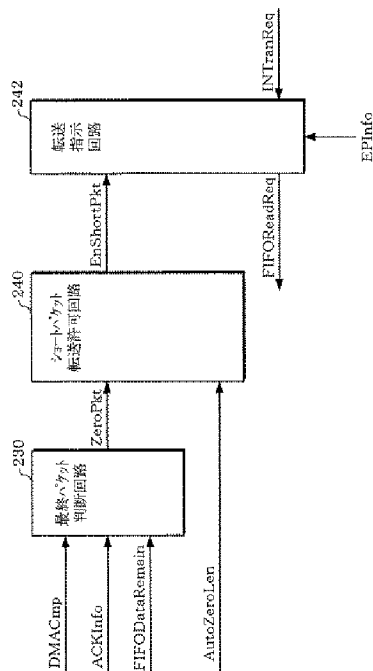
【図11】



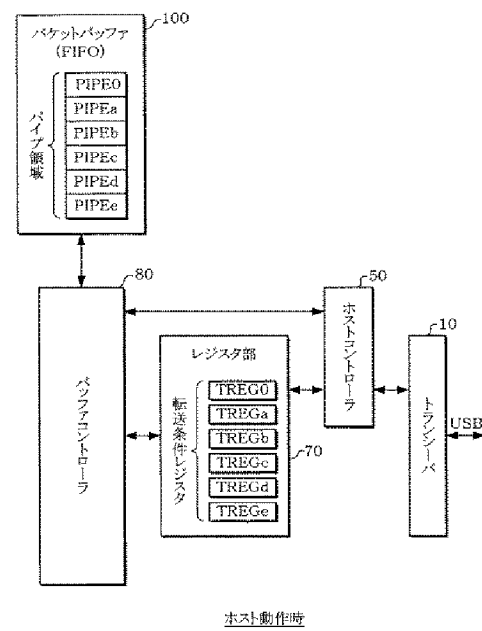
【図12】



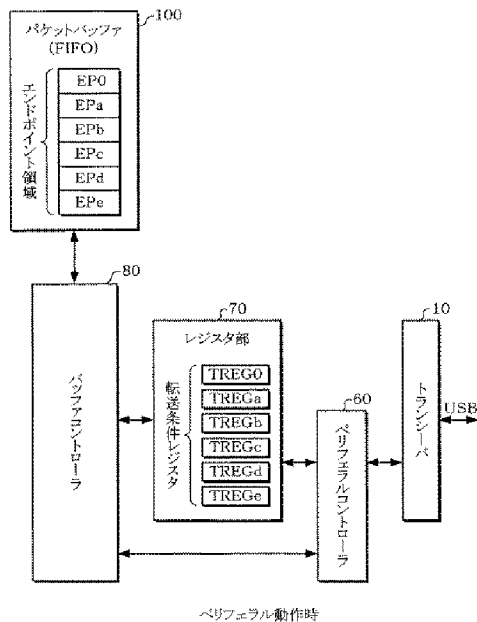
【図13】



【図14】



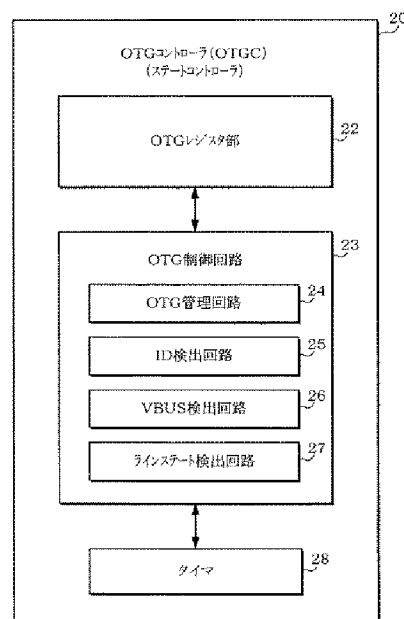
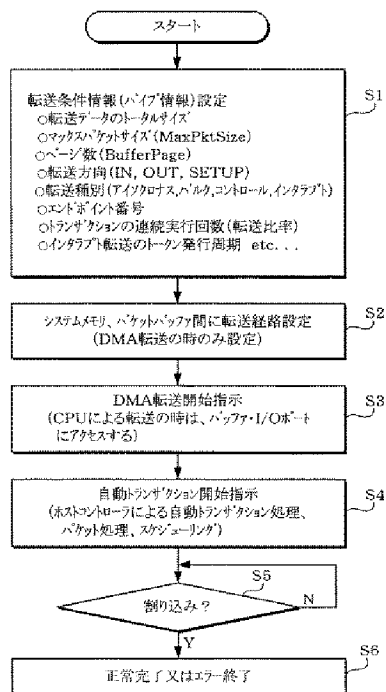
【例16】



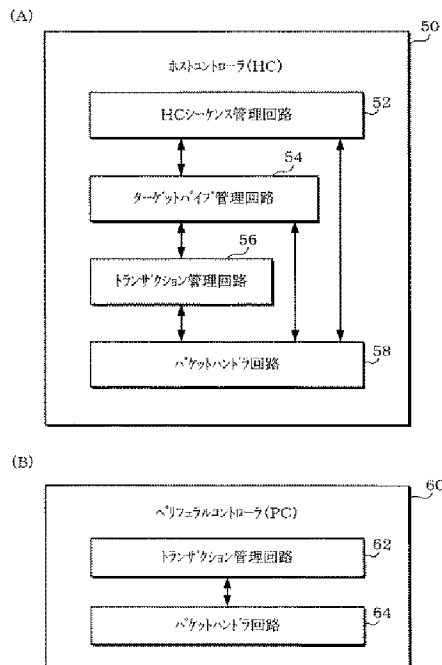
【例16】

|                                                           |                                 |                                      |                                                                                                  |
|-----------------------------------------------------------|---------------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------|
| 転送条件(シフト<br>方向)制御ビット<br>(O <sub>0</sub> ～O <sub>7</sub> ) | H/C(PCIE)/EP<br>共用レジスタ          | 各動作モードで、<br>ペリフェラルが要求する<br>共通されるレジスタ | データの転送方向、転送優先制。<br>ペリフェラル番号、データパス等。<br>ペリフェラDMA領域の有無等。                                           |
|                                                           | H/C(PD)用レジスタ                    | 各動作モードに依り、のみ<br>使用されるレジスタ            | インターホストのワーク、移行同期<br>レジスタ、高速データ受け取りフロッ<br>グレジスタ、転送データのエラー検出、<br>自動リトライなどの関係指示。<br>自動リトライ起動手元へ指示等。 |
|                                                           | PC(EP)用レジスタ                     | ペリフェラル動作時(EP)に<br>のみ使用されるレジスタ        | 互換性、インタープリ<br>ットシー指定等。                                                                           |
|                                                           | バックスピンドル(IPC)用の<br>共用レジスタを制御ビット | ペリフェラルがHPOのアクセス<br>制御を行なうためのレジスタ     | バックスピンドル、バックアップ、エンゲイ<br>メント、実行待ち状態等。                                                             |
| インターフェイス                                                  | インターフェイスレジスタ                    | 第1込みのインターフェイス(要因)<br>未定レジスタ          | OTCC系、HC(PE)伝送系、PC(EP)伝送等。                                                                       |
|                                                           | インターフェイスレジスタ                    | 第2込みのインターフェイス<br>アドレスを規定するレジスタ       | OTCC系、HC(PE)伝送系、PC(EP)伝送等。                                                                       |
|                                                           | プログラクエリレジスタ                     | プログラムで使用されるレジスタ                      | プログラクエリが指示等。                                                                                     |
|                                                           | プログラクエリレジスタ                     | Xen、OTCC、HC(EP)の<br>プログラム内で使用されるレジスタ | Xen制御、OTCCプログラム、<br>HCプログラム、プログラム番号                                                              |
| DMA制御レジスタ                                                 |                                 | DMA転送に関する<br>動作を行うレジスタ               | DMA転送開始指示、<br>DMA転送データのオフセット等。                                                                   |

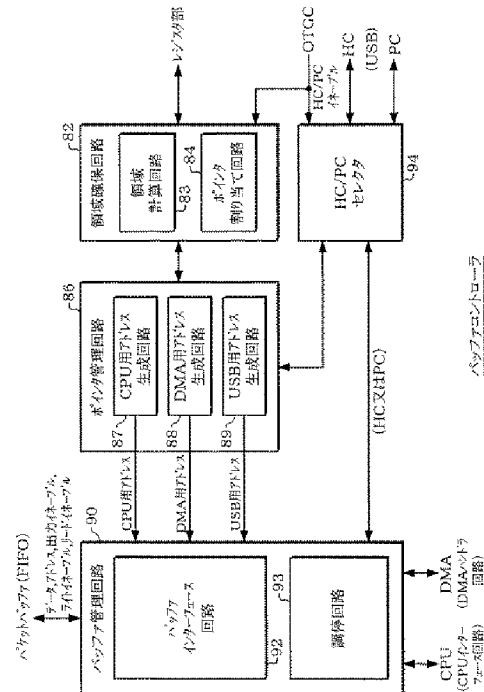
【例18】



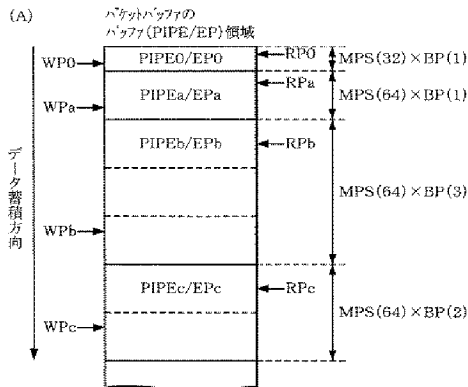
【図19】



【図20】



【図21】

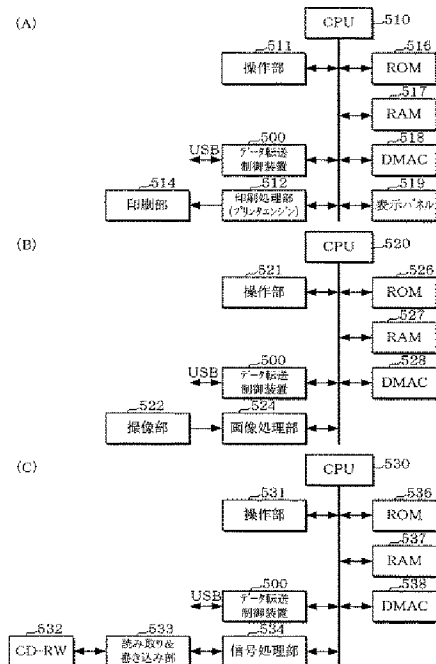


|    | DMA転送使用 | CPU転送使用 |
|----|---------|---------|
| WP | DMA     | CPU     |
| RP | USB     | USB     |

(C) 受信時 (USB→バグアコンントローラ→DMA又はCPU)

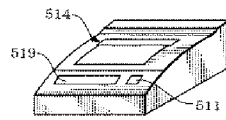
|    | DMA転送使用 | CPU転送使用 |
|----|---------|---------|
| WP | USB     | USB     |
| RP | DMA     | CPU     |

【図22】

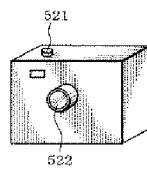


【図23】

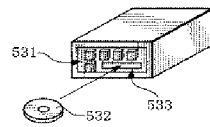
(A)



(B)



(C)





- (72)発明者 石田 卓也  
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
- (72)発明者 神原 義幸  
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
- (72)発明者 長尾 謙陽  
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
- F ターム(参考) 5B014 GA26 GC12 GC29  
5B061 FF00 RR03  
5B077 DD12